



Universität Ulm | 89069 Ulm | Germany

ulm university universität
uulm

**Fakultät für
Ingenieurwissenschaften
und Informatik**
Institut für Datenbanken und
Informationssysteme

MEDo: Mobile Technik und Prozessmanagement zur Optimierung des Aufgaben- managements im Kontext der klinischen Visite

Diplomarbeit an der Universität Ulm

Vorgelegt von

David Langer

mail@davidlanger.de

Gutachter

Prof. Dr. Manfred Reichert

Dr. Alena Hallerbach

Betreuer

Dipl. Inf. Rüdiger Pryss

2012

Fassung 22. März 2012

© 2012 David Langer

Dieses Werk bzw. Inhalt steht unter einer Creative Commons Namensnennung-Nicht-kommerziell-Weitergabe unter gleichen Bedingungen 3.0 Lizenz. Um eine Kopie dieser Lizenz anzusehen, besuchen Sie

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/> oder senden Sie einen Brief an Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Satz: X₃TeX

MEDo: Mobile Technik und Prozessmanagement zur Optimierung des
Aufgabenmanagements im Kontext der klinischen Visite

MEDo: Combining process-supported, mobile task management with patient data
to use on ward rounds

ZUSAMMENFASSUNG. Die Visite im Krankenhaus ist maßgebend für die Entscheidungsfindung an vielen Stationen. Während der Visite werden Aufgaben ermittelt und koordiniert. Diese werden allerdings nicht formal organisiert und sind durch Medienbrüche fehleranfällig. Ärzt/innen/e wissen nicht in welchem Status sich Behandlungen und Untersuchungen befinden.

In dieser Arbeit wurden Prozessmodelle für vier Visiten in unterschiedlichen Stationen erstellt und verglichen. Basierend auf den Gemeinsamkeiten wurde der Prototyp einer iPad-Applikation entwickelt und eine prozessunterstützte, mobile Aufgabenverwaltung mit Anschluss an Patientendaten realisiert. In Nutzerevaluationen und Usability-Tests hat sich gezeigt, dass alle Befragten das System einsetzen würden und sie die Geschwindigkeit der Aufgabenerstellung mit der von Stift und Papier gleichsetzen.

ABSTRACT. Medical rounds are highly influential in the decision-making-process at most wards. In the course of a round, tasks are determined and distributed. But they are not formally organised – instead they are jotted down and thus are prone to errors. Physicians are not notified about new results of tests they requested.

In this work I present and analyze process models for four medical rounds in different wards. Based on common features, I developed an iPad application which combines process-supported, mobile task management with patient data and information. User evaluation and usability tests then showed that all physicians would introduce such a system on their wards. Findings also show that they put the input speed on a level with that of pen and paper.

In conclusion, I could show that physicians can create, monitor and share tasks by using a mobile, user friendly platform.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziel	2
1.2	Entwicklungsschritte	3
2	Grundlagen	5
2.1	Visite im Krankenhaus	5
2.2	Prozessmanagement	7
2.2.1	IT-Unterstützung im Krankenhaus	9
2.3	Laufzettel als Pivot-Element der Visite	10
2.3.1	Probleme des Papier-Laufzettels	11
2.3.2	Vorteile eines digitalen Laufzettels	12
2.4	Tablets	13
3	Visite unter Prozessaspekten	17
3.1	Innere Medizin	18
3.2	Notaufnahme	21
3.3	Orthopädie	22
3.4	Chirurgie	25
3.5	Zusammenfassung	26
4	Anforderungsanalyse	29
4.1	Anforderungen an den Laufzettel	30
4.2	Anforderungen an die Hardware	33
4.3	Daten	34
4.3.1	Zugriffe	35
4.4	Szenarien	35
4.5	Zusammenfassung	35
5	Entwurf	37
5.1	Informationsarchitektur	38
5.2	Navigationskonzept	40
5.3	Screens	42
5.3.1	Stationsschirm	42
5.3.2	Aufgaben erstellen und anzeigen	42
5.3.3	Navigationsleiste	44

5.3.4	Patientenschirm	44
5.4	Styleguide: Piktogramme, Farben und visueller Stil	46
5.4.1	Typographie & Raster	49
5.4.2	Farben	49
5.4.3	Piktogramme	50
5.5	Zusammenfassung	50
6	Implementierung	53
6.1	Model	54
6.1.1	ER-Modell	55
6.2	View	57
6.2.1	MDTextModuleListControl	57
6.3	Controller	59
6.3.1	LoginViewController	61
6.3.2	OverViewController	63
6.3.3	PatientViewController	65
6.3.4	NewTaskViewController	71
6.3.5	TaskViewController	73
6.3.6	MDPortraitSplitViewController	73
6.4	Workflowunterstützung im System	76
6.4.1	XML-Datei	76
6.4.2	MDProcessManager	77
6.5	Zusammenfassung	78
7	Evaluierung	83
7.1	Ablauf	83
7.1.1	Erster Teil des Interviews	83
7.1.2	Usability-Test	83
7.1.3	Zweiter Teil des Interview	84
7.2	Ergebnisse	84
7.2.1	Erkenntnisse aus dem Usability-Test	85
7.2.2	Ergebnisse aus den geschlossenen Fragen	86
7.2.3	Ergebnisse des freien Teils des Interviews	86
7.3	Diskussion	89
8	Fazit	91
8.1	Ausblick	91
8.2	Schlusswort	93

Anhang	95
A Dokumente für die Datenerhebung	97
B Literaturverzeichnis	103
C Akronyme	109
D Abbildungsverzeichnis	111
E Tabellenverzeichnis	113

1

Einleitung

Ein Aufenthalt im Krankenhaus ist für Patienten anfangs ein Schock. Darauf folgt das Gefühl von Kontrollverlust: Der Patient als Laie versteht vieles von dem nicht, was um ihn herum passiert. Um dem entgegenzuwirken, braucht der Patient einen kompetenten Gesprächspartner.

Patientengespräche dienen damit nicht nur der Diagnostik, sondern fungieren – wenn sie mit entsprechender Sorgfalt als Arbeitsethos im Krankenhaus gepflegt werden – auch als Therapeutikum. Denn die „Wiederherstellung von Verstehbarkeit, Bewältigung, Bedeutsamkeit ist ein zentraler gesundheitsfördernder Faktor.“ [14, S. 317]. Ein Patient, der seine Leiden, deren Ursachen und Therapiemöglichkeiten kennt, ist das Idealbild des selbstständigen und mündigen Patienten [19]. Es obliegt den Ärzt/inn/en und ihren Kommunikationsfähigkeiten, den Patienten so zu informieren, dass dieser kompetent beraten wird. Ein informierter Patient kann besser mit der ungewohnten Situation des Krankenhausaufenthaltes umgehen.

Um diese Patientenkommunikation zu gewährleisten, werden in den meisten Stationen zweimal am Tag Visiten durchgeführt. Dabei gehen Ärzt/innen/e zusammen mit dem Pflegepersonal von Patient zu Patient und besprechen mit ihnen den aktuellen Stand, eventuelle Untersuchungsergebnisse und das weitere Vorgehen.

Bereits im 18. Jahrhundert wurden Visiten dieser Art populär [15]. Zur damaligen Zeit gab es keine Untersuchungsergebnisse, die zur Visite mitgeführt werden konnten. Mit dem Voranschreiten der Wissenschaft und Technik standen allerdings immer mehr Daten und Informationen über Patienten zur Verfügung. Diese Daten sollten natürlich während der Visite ebenfalls zur Verfügung stehen. Dazu wurde früher ein mit Patientenakten bestückter Visitenwagen eingesetzt.

Heutzutage wird verstärkt versucht, Patientendaten digital zu halten. So sollen Kosten und Zeit gespart werden. Kosten können durch Ausschluß von Doppeluntersuchungen und Fehlbefundungen, aufgrund von fehlenden Informationen, eingespart werden [38]. Dadurch, dass Daten sofort digital verfügbar sind, spielen Versandzeiten von Dokumenten in der Hauspost keine Rolle mehr. Die Kommunikation via E-Mail ist beispielsweise in vielen Krankenhäusern mittlerweile üblich.

Patientenakten setzen sich aus Informationen aus verschiedenen Quellen, wie Röntgenstationen und externen Ärzt/inn/en, zusammen. Meist sind nicht alle Ein-

zelkomponenten umgestellt sind, weswegen oft trotzdem noch Papierakten existieren. Viele Stationen und Kliniken befinden sich in einem Stadium zwischen analoger und digitaler Patientenakte.

Aus diesem Grund muss auch bei der Visite umgedacht werden. Beispielsweise werden relevante Elemente der Akte ausgedruckt und mitgenommen. Eine andere Lösung besteht darin, den Visitenwagen mit einem Laptop auszurüsten oder einen Tablet-Computer mit auf Visite zu nehmen. Diese Techniken gehören zu den Inhalten der *mobilen Visite*: Im einfachsten Fall wird es dem Arzt oder der Ärztin ermöglicht, mithilfe eines mobilen Endgeräts auf das Krankenhausinformationssystem (KIS) zuzugreifen. In der Realität ist die Umsetzung allerdings aufgrund verschiedener Altsysteme, inkompatiblen Schnittstellen und Kostenbedenken kompliziert und in wenigen Krankenhäusern anzutreffen.

Aus der Visite ergeben sich viele Aufgaben, wie Untersuchungsanforderungen und Therapiemaßnahmen. Da diese Aufgaben allerdings oft verlangen, dass Formulare ausgefüllt werden, können sie meist nicht direkt am Patientenbett erledigt werden. Stattdessen notieren entweder die anordnenden Ärzt/innen/e oder Assistenzärzt/innen/e die Aufgaben auf einem Papierblatt.

Zwischen dem papierbasierten Aufgabenmanagement auf der einen und der Digitalisierung der desktopgebundenen Patientenakte auf der anderen Seite entwickelt sich ein Bruch: Aufgaben werden durch Medienwechsel fehleranfällig und sind nicht nachverfolgbar.

Prozesse in der Klinikumgebung sind oft nicht formal erfasst, und damit weder überprüfbar, noch können sie optimiert werden. So sind Workflow-Management-System (WfMS) ebenfalls nicht weit verbreitet. Insbesondere sind Prozesse, die bei der Diagnose und Therapie des Patienten eine Rolle spielen, dynamisch [13] und stellen somit eine Herausforderung an die technische Prozessunterstützung (Workflows) dar.

MEDo (MedicalDo), das einerseits Aufgaben digital verwaltet und andererseits Patientendaten anzeigt, kann diesen Bruch überwinden. Gleichzeitig werden durch die Einbindung von Prozessen in die digitale Arbeitsverwaltung potenziell Fehlerquellen ausgeschlossen, Prozesszeiten verbessert und Statusabfragen ermöglicht. Der Schlüssel dabei liegt darin, dass die Applikation auf einem Tablet entwickelt wird und somit mobil während der Visite mitgenommen werden kann.

1.1 Ziel

Das Ziel dieser Diplomarbeit ist herauszuarbeiten, wie während der Visite durch mobile Prozessunterstützung Arbeit und Zeit gespart werden können. Es soll ein Prototyp entwickelt und von Ärzt/inn/en evaluiert werden.

Um entscheiden zu können, wo die Unterstützung anzusetzen ist, sollen mehrere Visiten beobachtet und aus Sicht des Prozessmanagements analysiert werden. Im

Anschluß daran soll ausgewertet werden, ob und wie Prozesse bei der Visite unterstützt werden können. Es ist außerdem zu überprüfen, an welchen Punkten dort eine mobile Unterstützung einen Gewinn bieten kann.

Das bisher rein papierbasierte und informelle Aufgabenmanagement soll durch ein mobiles System unterstützt werden. Dabei sollen die Peripherieprozesse¹, die im ersten Teilziel identifiziert wurden, durch Workflows nachverfolgbar unterstützt werden. Patienteninformationen müssen ebenfalls erreichbar sein. Dabei sollen die Arbeitsweisen der Ärzte auf dem mobilen Gerät weiterhin unterstützt werden. Das System ist als Prototyp auf einem iPad zu entwickeln.

Großes Augenmerk soll dabei auf die Gestaltung der Benutzerschnittstelle gelegt werden, da diese entscheidend für die Akzeptanz der Benutzer ist. Der Prototyp soll schließlich evaluiert werden, um zu überprüfen, ob und wie eine solche Hilfe von den Ärzten angenommen wird.

1.2 Entwicklungsschritte

Durch die Ungewissheit, wie das System letztendlich eingesetzt wird und die Fokussierung auf das Benutzerinterface, habe ich mich für einen Ansatz nach dem *exploratory prototyping* [9] entschieden. Auf die Anwendung eines kompletten Software-Entwicklungsmodells wurde in Anbetracht des Prototypcharakters verzichtet. Stattdessen sind Techniken aus dem Gebiet des *Agile Modelling* zum Einsatz gekommen. Dementsprechend habe ich die Entwicklung entlang des *UI prototyping process* [3] geplant, der in Abbildung 1.2 auf der nächsten Seite zu sehen ist.

Im Folgenden soll nun der Verlauf der Arbeit, der in Abbildung 1.1 grafisch dargestellt ist, erklärt werden.

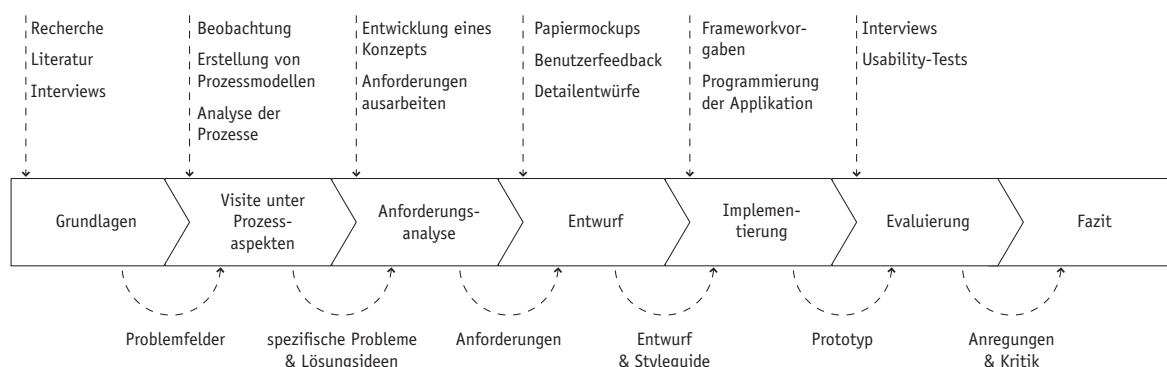


Abbildung 1.1: Übersicht über den Aufbau der Arbeit

Um mit dem Prototypen auch die aktuellen und eingespielten Vorgehensweisen der Ärzte zu unterstützen, führte ich zu Beginn intensive Beobachtungen der Ar-

¹Peripherieprozesse: Prozesse, die an den Ablauf der Visite angeschlossen sind.

beitsweise bei der Visite durch. Die entstandenen Erkenntnisse wurden in darauffolgenden Interviews hinterfragt und verifiziert. Um die Gemeinsamkeiten herauszuarbeiten, wurden Prozessmodelle aller beobachteten Visiten angefertigt. Die Analyse ergab, dass wegen der Komplexität und Dynamik nicht die gesamte Visite durch ein Modell abgedeckt und unterstützt werden kann.

Wohl aber ließen sich Teilprozesse erkennen, die durch die richtige Unterstützung den Arbeitsfluss erleichtern. Aus den Beobachtungen und Prozessen wurden schließlich Anforderungen an das System abgeleitet und durch Gespräche mit Ärzt/inn/en abgesichert.

Die Entwurfsphase begann mit der Entscheidung für die Entwicklung auf einem iPad. Dadurch wurde zum einen die verfügbare Größe für die Benutzerschnittstelle festgelegt, zum anderen gibt die Entscheidung für eine Plattform auch Designguidelines wie die iOS Human Interface Guidelines [6] vor. Nach der Festlegung der benötigten Daten habe ich mithilfe von vielen, möglichst verschiedenen Papierprototypen ein finales Konzept erstellt. Dies wurde anschließend in Objective-C programmiert.

Den entstandenen Prototyp habe ich schließlich zusammen mit Ärzt/innen/e evaluiert. Die im Verlauf dieser Auswertung herausgearbeiteten Anregungen und Kritikpunkte werden im Ausblick vorgestellt.

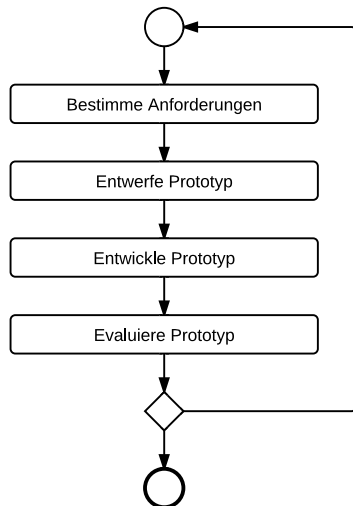


Abbildung 1.2: UI prototyping process

2

Grundlagen

Dieses Kapitel soll die Grundlagen zum Verstehen des Kerns dieser Arbeit schaffen. Dazu werden die Begriffe „Visite“ und „Laufzettel“ definiert, sowie das Aufgabenmanagement in diesem Kontext. Es wird außerdem das Prozessmanagement und dessen Anwendung im Klinikumfeld besprochen und eine Einführung in die Thematik der Tablets gegeben.

2.1 Visite im Krankenhaus

Visiten sind „in der Regel die einzigen Zeitpunkte, an denen ein Team mit den Patienten zusammenkommt, und sich über den aktuellen Stand der Versorgung und Genesung des Patienten und damit über den Stand der ‚primary task‘ Patientenversorgung informiert und austauscht“ [56]. Sie werden ausserdem durchgeführt, damit bei der Schichtübergabe Informationen weitergegeben werden können und so das nächste Team nahtlos weiterarbeiten kann. Weiterhin besteht auch von Seiten der Chefärzte/innen und Stationsleit/er/innen einen Überblick über die Stationen, für die sie verantwortlich sind, zu bekommen. Sie führen meistens einmal pro Woche eine eigene Visite durch. Es gibt also im Verlauf der Woche Visiten mit unterschiedlichen Zielsetzungen.

Ebenso gibt es von Seiten der Patienten Motive für die Visite. Sie bekommen durch den direkten Kontakt regelmäßig die neuesten Informationen über ihren Genesungsverlauf. Außerdem werden sie über Abläufe, die geplanten Untersuchungen und Therapiemaßnahmen aufgeklärt und haben die Möglichkeit Fragen zu stellen.

Obwohl die Motive hinter den Visiten ähnlich sind, führt sie doch jede Station anders durch. Der Ablauf ist abhängig von der Art der Station, den Anforderungen an die Visite (Schichtübergabe, tägliches Bild), der Liegezeit und Anzahl der Patienten und den Präferenzen der Oberärzt/innen/e.

Während der Visite machen sich die Ärzt/innen/e ein Bild vom Krankheits- und Genesungsverlauf des Patienten. Daraus entstehen nach Absprache Entscheidungen über den Behandlungsplan [59]. Dazu ziehen die beteiligten Ärzt/innen/e verschiedene Informationsquellen zu Rate: Neben der physischen Untersuchung beispielsweise von Wunden und Verletzungen, werden Untersuchungsergebnisse aus

der bildgebenden Diagnostik und Laborbefunde hinzugezogen. Genauso wichtig sind die Gespräche mit den Patienten sowie die Eindrücke und die Beobachtungen des Pflegepersonals. Aus diesen Informationen ergeben sich die nächsten Therapieschritte und Untersuchungen.

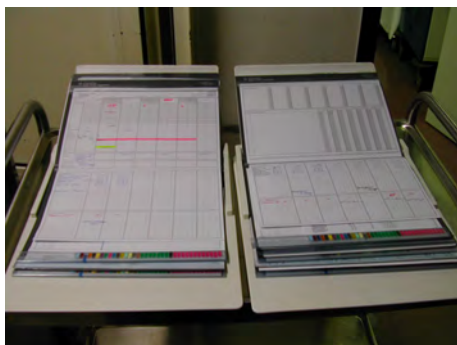


Abbildung 2.1: Mappen für die Pflegedokumentation [36]

Bei der Visite wird meist die Pflegedokumentation mitgenommen (siehe Abbildung 2.1). Dabei handelt es sich um eine Sammlung von verschiedenen Dokumenten, die in der Gesamtheit alle Informationen über den Pflegeprozess ergeben. Sie enthält die Patientenkurve, in der verschiedene Vitaldaten des Patienten wie Herzfrequenz, Blutdruck und Körpertemperatur dokumentiert werden. Ebenso wird in der Pflegedokumentation die Medikation protokolliert und geplant. Ärzt/innen/e setzen diese fest und zeichnen dies und Änderungen ab. Ziel ist, dass Änderungen an der Medikation nachvollziehbar sind.

In der Regel lassen sich Visiten in die folgenden vier Phasen einteilen [51]:

Vorbereitung Im Vorfeld der Visite stellen alle Ärzte die relevanten Informationen über ihre Patienten zusammen. Beispielsweise drucken sie Röntgenbilder aus oder legen die Pflegedokumentation bereit.

Visite Ante Portas Vor Betreten eines jeden Patientenzimmers stellt einer der Ärzte den Patienten den anderen Teilnehmern der Visite vor. Gegebenenfalls werden bereits erste Anordnungen gegeben.

Visite im Patientenzimmer Nach einer eventuellen physischen Untersuchung des Patienten werden ihm nun die Befunde und der weitere Behandlungsplan erläutert; Untersuchungen und Therapiemaßnahmen werden geplant.

Nachbereitung Nach dem die beiden vorangegangenen Punkte für jeden Patienten wiederholt wurden, werden Pläne für den weiteren Verlauf abgearbeitet, also beispielsweise Untersuchungsanordnungen ausgearbeitet, unterschrieben und verschickt.

Früher wurden bei den Visiten sogenannte Visitenwägen eingesetzt. Diese Wägen bestückte man vor jeder Visite mit den entsprechenden Patientenakten. Während der Visite kümmerte sich eine Ärztin/ein Arzt darum, die benötigten Akte herauszusuchen und zu verteilen. Mittlerweile kommt diese Art der Visite immer seltener zum Einsatz, da die Daten elektronisch gespeichert werden. Die meisten Krankenhäuser sind schon so weit digital integriert, dass Röntgenbilder und Laborbefunde primär digital einsehbar sind.

Daraus ergibt sich, dass der vorrangige Zugriff auf diese Daten durch ein desktopgebundenes Krankenhausinformationssystem (KIS) geschieht. Manche Stationen verwenden einen digitalen Visitenwagen, der, durch einen darauf angebrachten Laptop

oder Desktop-Rechner, auch während der Visite Zugriff auf die Daten erlaubt. Eine Alternative wäre zum einen, die Visite vorzubereiten und relevante Teile der Patientenakte wie zum Beispiel Röntgenaufnahmen oder Laborbefunde auszudrucken und mitzunehmen. Zum anderen kann die Visite in „Datensichtung am Desktop-PC“ und „Patientengespräch am Bett“ geteilt werden. Wenn kein Patientenkontakt bei der Visite zustande kommt, redet man auch von einer Kurvenvisite.

Nach der Visite müssen die angefallenen Aufgaben bearbeitet werden. Dabei muss viel Zeit (etwa 33,7% bis 40,6% [20, S. 4], bzw. ca. 3 Stunden pro Tag [11]) für die Dokumentation der Maßnahmen, Befunde und Untersuchungen aufgewendet werden.

Ständige Unterbrechungen wie Telefonanrufe machen es den Ärzt/innen/e schwer, sich durchgängig auf die Visite zu konzentrieren. Abhängig von der Länge der Visite und der Anzahl der Teilnehmer erhöht sich die Wahrscheinlichkeit einer Unterbrechung.

Für eine patientenzentrierte Visite ist die Verfügbarkeit möglichst aller Patientendaten am Bett erforderlich. Da dies wegen desktopgebundenen, digitalen Patientenakten nicht gegeben ist, gibt es Entwicklungen, die sich unter dem Begriff *mobile Visite* zusammenfassen lassen [4, 28, 58]. Diese umfasst mobile Clients, wie Visitenwägen mit Desktop-PCs oder Notebooks und erste Tablet-Lösungen. Meist ist die verwendete Software nicht speziell an die Möglichkeiten und Beschränkungen der Hardwareplattform angepasst. Gerade bei Tablets, die mittels direkter Manipulation auf einem berührungsempfindlichen Bildschirm gesteuert werden, ist aber die Anpassung sehr wichtig.

Trotz der Entwicklungen in Richtung einer volldigitalen Patientenakte [19] greifen Ärzt/innen/e auf einfache Laufzettel zur Aufgabenverwaltung zurück. Ein Laufzettel ist eine Art handschriftlich geführte Arbeitsliste, auf der Aufgaben vermerkt und nach der Durchführung gestrichen werden. Die Aufgaben werden hauptsächlich während der Visite erfasst und im weiteren Verlauf des Tages erledigt. Somit ist dieses Papier die zentrale Anlaufstelle für alle Ärzte einer Station, um zu erfassen, was die Aufgaben des Tages sind.

2.2 Prozessmanagement

Der Begriff „Prozess“ beschreibt einen Ablauf von Bearbeitungsschritten [22, S. 7]. Er grenzt sich von einem Projekt dadurch ab, dass ein Prozess wiederholt wird. Er ist als ein Plan zu verstehen – erst wenn er ausgeführt wird, spricht man von einer Prozessinstanz. Prozesse werden von Georgakopoulos u. a. [24] in drei Kategorien eingeteilt:

Materielle Prozesse bezeichnen Vorgänge bei denen am Ende ein Produkt steht. Das bedeutet auch, dass die meisten Aufgaben von Personen physisch ausgeführt werden.

Informationsprozesse sind teil- sowie komplett automatisierte Abläufe, bei denen Personen die Aufgaben am Computer ausführen. Basis dafür sind Technologien wie Datenbanken, Transaktionsabwicklung und verteilte Systeme.

Geschäftsprozesse bezeichnen Vertriebsvorgänge in Organisation, die sich meist aus den oben genannten materiellen und Informationsprozessen zusammensetzen.

Workflows beschreiben die operativ-technischen Schritte, die für die Ausführung eines Geschäftsprozesses nötig sind. Der Unterschied zu Geschäftsprozessen ist damit die Eindeutigkeit der Aufgabenschritte, die zur Ausführung nötig ist. Insbesondere spezifizieren Workflows die Mensch-Maschine-Interaktion mit dem System. Es existieren formale Beschreibungssprachen [45] für Workflows, welche die Ausführung durch ein Computersystem, ein sogenanntes Workflow-Management-System (WfMS), unterstützen.

Anstatt Aufgaben einzelnen Personen zuzuweisen, werden Benutzerrollen zur Abstraktion eingesetzt. Eine Rolle beschreibt dabei gemeinsame Eigenschaften und Rechte von Benutzern. So kann leichter mit Personalveränderungen umgegangen werden.

In einer sogenannten Arbeitsliste bekommen Benutzer eines WfMS anstehende Aufgaben angezeigt. Sie listet alle Aufgaben, die für die Rolle des angemeldeten Benutzers zur Verfügung stehen, auf. Um die Bearbeitung zu beginnen, wählt er eine aus, markiert sie als „in Bearbeitung“ und beginnt dann den Arbeitsschritt auszuführen. Hat er die Aufgabe abgeschlossen, markiert er sie als erledigt.

Zur Darstellung von Geschäftsprozessen und Workflows gibt es verschiedene Formalismen, wie beispielsweise die Business Process Modeling Notation (BPMN) [50, 61], Ereignisgesteuerte Prozessketten (EPK) [23, 40], Petri-Netze [1, 2, 42] oder Flussdiagramme [17, 32]. Besonders hervorzuheben ist dabei BPMN: Ein von der Object Management Group (OMG) verabschiedeter Standard, welcher sowohl für die Business-Seite (Geschäftsprozesse), als auch die IT-Seite (Workflows) eine einheitliche Symbolpalette bietet [41, 43]. Abbildung 2.2 zeigt einen einfachen Prozess in BPMN.

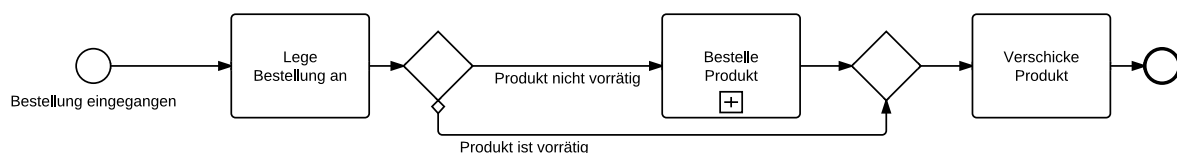


Abbildung 2.2: Ein Prozessbeispiel in BPM-Notation. Nach dem Startereignis (*event*) am linken Bildrand folgt eine Aktivität (*activity*), verbunden durch eine Kante. Danach steht eine datenabhängige Entscheidung (*gateway*) an. Durch den kleinen Rhombus an dessen unterer Kante wird der Standardfall gekennzeichnet. Das Plussymbol kennzeichnet einen Subprozess und der kräftigere Kreis das Endereignis.

Um die Übersichtlichkeit bei der Darstellung zu verbessern, werden oft Subprozesse eingesetzt, die Verhalten und Abläufe ausblenden. Auf diese Weise kann ein Prozess, beispielsweise für das Management, abstrahiert dargestellt werden. Für die genauere Ausführung müssen die Prozessschritte weiter verfeinert werden.

Die Gründe eines Unternehmens, die eigenen Prozesse zu optimieren, sind vielfältig: Sei es die Effizienz bei der Produktion erhöhen, Kundenzufriedenheit verbessern, Qualität von Produkten respektive Services steigern oder Kosten reduzieren.

Sollen in einem Unternehmen Workflows eingesetzt werden, müssen zuerst genaue Ziele für die Optimierung definiert werden. Danach werden Prozesse identifiziert und modelliert. Aufgrund dieser Beschreibung aus Geschäftssicht wird nun ein Workflow modelliert, der den Prozess um Details verfeinert. Anschließend wird der Workflow implementiert, also in einem WfMS zur Ausführung gebracht. Schließlich muss der Prozess beobachtet werden, um die daraus gewonnenen Daten mit den Zielen vergleichen zu können. Ausgehend von diesen Ergebnissen, sind nun weitere Änderungen am Prozessmodell möglich [45]. Diese Vorgehensweise wird *process management lifecycle* genannt (siehe Abbildung 2.3).

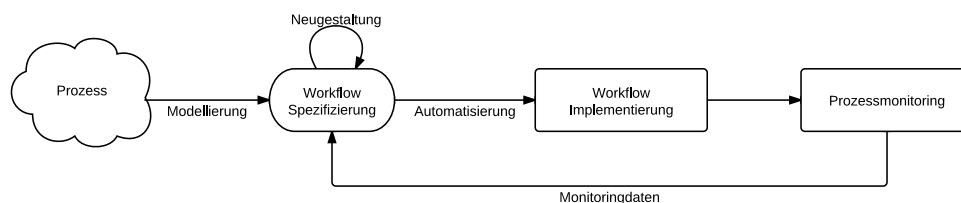


Abbildung 2.3: *Process mangagement lifecycle*

Die Einführung von Prozessen und insbesondere von WfMS in Unternehmen ist oft nicht einfach. Zum einen sind technische Hürden zu überwinden, zum anderen müssen die Mitarbeiter eingebunden werden. Dies kann schwierig sein, da mögliche Konsequenzen von Prozessoptimierungen Verringerung von menschlichen Aktivitäten – und damit Entlassungen – sein können. Es ist deshalb wichtig, dass „die Bereitschaft und das Verständnis der Prozessbeteiligten, die neuen Strukturen zu verstehen, zu akzeptieren und umzusetzen,“ [22, S. 3] gestärkt wird.

2.2.1 IT-Unterstützung im Krankenhaus

Im Krankenhaus sind zwei Kategorien von Prozessen zu unterscheiden. Zum einen sind Geschäftsprozesse vorzufinden, die Vorgänge wie Abrechnungen, Dokumentation und Urlaubsanträge abdecken. Dies sind Vorgänge, die es auch in jeder anderen Organisation gibt. Sie können unproblematisch mithilfe von WfMS unterstützt werden.

Zum anderen existieren Prozesse, bei denen der Patient und dessen Therapie im Vordergrund steht. Die prozesstechnische Unterstützung ist hier noch am Anfang [34]. Beispielsweise sollen Decision Support Systems (DSS) Ärzt/inn/en bei der

Suche nach Entscheidungen unterstützen [27]. Entscheidungen müssen aufgrund von Daten, beispielsweise Laborwerten oder CT-Aufnahmen, anhand des spezifischen medizinischen Wissens der Ärzt/innen/e und unter Beachtung des aktuellen Zustands des Patienten getroffen werden. Die Anzahl dieser Faktoren machen es im Einzelfall schwer, die richtige Entscheidung zu treffen. DSS könnten beispielsweise den aktuellen Fall automatisch mit anderen, ähnlichen Fällen unter Beachtung der oben genannten Faktoren vergleichen, um so den Ärzt/inn/en Vorschläge zu unterbreiten. Dazu müssen existierende medizinische Leitfäden maschineninterpretierbar aufbereitet werden und an Patienteninformationen angebunden werden.

Diese werden im Krankenhaus im Patientendatenbankmanagementsystem (PD-MS) gespeichert. Zusammen mit Kommunikations- und Workflow-Funktionen bildet es das Krankenhausinformationssystem (KIS) [52]. Solche Softwareprodukte versprechen, alle Daten der Patienten einheitlich in einem System zu speichern.

Diese Vision konnte bisher nicht in die Praxis umgesetzt werden. Viele Systeme, wie beispielsweise Röntgenabteilungen oder Labore, sind weiterhin nicht in das Gesamtsystem integriert. Die Daten sind in heterogenen Systemen verteilt, nur die funktionsbezogenen Zugriffe ermöglichen. Sie sind meist für die Organisationseinheiten ausgerichtet, in denen sie eingesetzt werden (Röntgenabteilung, Labor, Innere Medizin etc.).

So kann keine ausreichende Unterstützung von bereichsübergreifenden Prozessabläufen, die den Patienten im Fokus haben, stattfinden [44]. Diese Heterogenität der Systeme im Krankenhaus zu überwinden, ist eine Hauptaufgabe der medizinischen IT in der Zukunft [34].

In den Fokus der Öffentlichkeit rückt die krankenhausesübergreifende elektronische Patientenakten (EPA) in Deutschland in Form der elektronische Gesundheitskarte (EGK) [19]. Sie soll die zentrale Speicherung medizinischer Daten ermöglichen, bei der die Karte als Schlüssel zu diesen Daten dient. Man verspricht sich durch die einheitliche Führung einer elektronischen Akte pro Bürger Qualitätsverbesserungen, Rationalisierungen, bessere Wirtschaftlichkeit und die Stärkung der Patientenautonomie [16].

Nachdem die Einführung jahrelang aufgrund von technischen Problemen und dem Protest vieler Ärzte verzögert wurde [19], wird die EGK seit dem 1. Oktober 2011 als Versichertenkarte anerkannt. Noch sind allerdings keine der geplanten Funktionen, die einen Austausch von Informationen ermöglichen, implementiert [25]. Insbesondere der Datensicherheit misstrauen viele Versicherte [26].

2.3 Laufzettel als Pivot-Element der Visite

Anstehende Behandlungs- und Untersuchungsschritte müssen zeitversetzt und über den Rest des Tages verteilt ausgeführt werden. Ein ständiger Begleiter der Ärzt/innen/e ist somit ein Laufzettel, der während der Visite ausgefüllt und im Laufe des

Tages abgearbeitet wird. Um sich an anstehende Aufgaben zu erinnern, reicht es Patientennamen und Text aufzuschreiben. Sobald eine Aufgabe durchgeführt wurde, wird die entsprechende Zeile durchgestrichen.

Im Folgenden ein Auszug der Einträge, die auf Laufzetteln notiert werden:

- Erinnerungen an Rücksprachen mit Ober- oder Chefärzt/inn/en.
- Bei externen Ärzt/inn/en per Telefon nach Konsilen fragen und ob bereits Ergebnisse bzw. Rückmeldung vorhanden sind.
- Überprüfen, ob Ergebnisse von Untersuchungen eingetroffen sind.
- Erinnerungen, den Patienten in Richtung einer vermuteten Krankheit zu untersuchen.
- Planung der durchzuführenden Konsile.
- Langzeiterinnerungen (Beispiel: Sobald ein Patient von MRSA geheilt ist, soll er einer Röntgenuntersuchung unterzogen werden)

Manche Aktivitäten, wie beispielsweise die Anforderung einer Röntgenuntersuchung sind schnell erledigt. Teilweise besteht die Möglichkeit, dies während der Visite, mithilfe eines Tablets abzuschließen. Andere Aufgaben, wie die Rücksprache mit Ober- oder Chefärzt/inn/en oder das Warten auf Ergebnisse eines Konsils können sich über mehrere Tage erstrecken. Noch langlebiger sind Erinnerungseinträge, wie eine Untersuchung, die erst möglich ist, nachdem der Patient eine noch bestehende Krankheit überwunden hat.

Die meisten Eintragungen werden während der Visite, also unter Zeitdruck, erstellt. Auf Papier kann der Benutzer jede Möglichkeit nutzen, Zeit zu sparen. Er kann sehr schnell schreiben, Einträge kennzeichnen, Pfeile und Verbindungsstriche zwischen Einträgen zeichnen und als wichtigstes Element: Details auslassen. Diese *Nicht-Spezifizierung* macht Stift und Papier zu einem hocheffizienten Werkzeug. Elektronische Werkzeuge fordern vom Benutzer spezifisch zu sein und sich festzulegen.

Teilweise nutzen Stationsärzt/innen/e auf einer Station eine gemeinsame Arbeitsliste. In diesem Falle liegt sie im Dienstzimmer aus. Hat einer der Ärzt/innen/e etwas freie Zeit, kann er ausgehend von den noch offenen Aktivitäten, die nächste Aufgabe in Angriff nehmen.

2.3.1 Probleme des Papier-Laufzettels

Handschriftlich geführte Laufzettel können zwar flexibel eingesetzt werden, sind allerdings mit Mängeln hinsichtlich der weiteren Benutzung, wie Wartung und Abarbeitung, belegt.

So bedeutet die Nutzung eines handschriftlichen Laufzettels, dass mindestens zweimal ein Wechsel zwischen Medien erfolgen muss. Zum einen bei der Erfassung, zum anderen bei der Ausführung. Der erste Wechsel ist besonders dann kritisch, wenn eine andere Person mitschreibt, beispielsweise Assistenzärzt/innen/e

für Oberärzt/innen/e. Der zweite Wechsel stellt ein Problem dar, wenn zwischen Aufgabenerstellung und Ausführung merklich Zeit vergangen ist oder nicht der Aufgabenersteller selbst die Aufgabe ausführt. Hier können Informationen verloren gehen.

Ist das Ende einer Schicht oder eines Tages erreicht, wird der Laufzettel von einer Ärztin/einem Arzt ins Reine geschrieben, also nicht bearbeitete Einträge auf eine neue Seite übertragen. Dies schafft einen sauberen Ausgangspunkt für den nächsten Tag, ist aber zeitaufwendig.

Weiterhin liegt der Laufzettel meist zentral im Dienstzimmer der Ärzte und ist damit nicht erreichbar für Ärzte, die beispielsweise in der Notaufnahme Dienst haben. Diese können dadurch freie Zeiten nicht nutzen, um z.B. Telefonate zu führen. So müssen die Ärzt/innen/e immer wieder zurück ins Dienstzimmer gehen, um zu sehen, welche Aufgaben noch offen sind.

Handschriftliche Notizen bergen zudem die Gefahr der Unleserlichkeit. Dies stört besonders dann, wenn mehrere Personen zusammen einen Laufzettel bearbeiten. Falsch verstandene Informationen oder missverständliche Anweisungen erschweren die Zusammenarbeit mit anderen an einem Dokument zusätzlich.

Ohne Archivierung der Aufgaben ist keine Rückverfolgung möglich. Interessant dabei ist, *welche Person wann welche Aufgabe ausgeführt hat*. Besonders bei Problemen oder Unklarheiten ist die Rückverfolgung von Arbeitsschritten sinnvoll. Aber auch bei der täglichen Arbeit ist es wichtig zu erfahren, ob eine Aufgabe bereits erfüllt wurde.

2.3.2 Vorteile eines digitalen Laufzettels



Abbildung 2.4: Applikation Erinnerungen. Bestandteil von iOS 5

Durch die Entwicklung der Krankenhausinformationsstruktur lässt sich ableiten, dass die Entwicklung in Richtung der elektronischen Akten geht. In absehbarer Zeit wird die gesamte Patientenakte in elektronischer Form geführt. Dazu sind neue Konzepte und Paradigmenwechsel beim Aufgabenmanagement nötig, wie die vorausgegangene Betrachtung zeigt.

Überträgt man das Konzept eines Laufzettels auf ein Smartphone, erhält man zunächst eine To-Do-Listen-Applikation. Auf dem iPhone ist beispielsweise bereits „Erinnerungen“ vorinstalliert (siehe Abbildung 2.4). Im Krankenhauskontext sind allerdings weitere Verbesserungen denkbar.

Beispielsweise könnte ein WfMS zusätzlich in den Laufzettel integriert werden. Auf diese Weise wären Einträge der Arbeitsliste des WfMS und *normale* Aufgaben gemischt. Die Benutzer könnten auf diese Weise ihre bisherige Arbeitsweise mit einem Laufzettel weiternutzen. Gleichzeitig können die Vorteile eines Prozesssystems genutzt werden. Es könnten Merkmale und Richtwerte, wie Zeiten oder Fehler erhoben werden.

Durch gezielte Änderungen an den Prozessen, sollte dann versucht werden, diese Werte positiv zu beeinflussen.

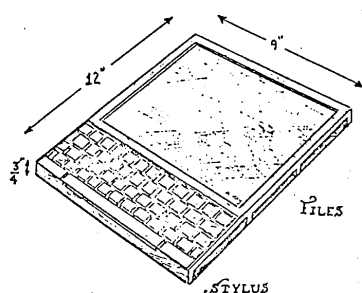
Durch WfMS sind Ergebnisse und Eingaben eines Anwenders sofort für die nächsten Anwender verfügbar. Ein Beispiel ist der Prozess „Blutuntersuchung“ in Abbildung 3.8 auf Seite 27. Wenn dieser digital unterstützt wird, kann die Pflegekraft direkt, während sie am Patientenbett steht, darüber informiert werden, dass eine Blutuntersuchung eines Patienten gewünscht wird. Sobald sie alles vorbereitet hat, bestätigt sie den Schritt im System. So wird eine Ärztin/ein Arzt informiert und kann Blut abnehmen. Sie/Er wird dann wiederum automatisch über den Status der Untersuchung informiert, sowie wenn das Ergebnis der Untersuchung verfügbar ist. Auf diese Weise werden Kommunikationswege verkürzt und die Gefahr, dass Informationen verloren gehen, wird verringert.

Unerlässlich ist, neben der Prozessunterstützung, auch die Integration in bestehende KIS. So hat der Benutzer nicht nur Aufgaben, sondern auch sofort alle Daten, die er zur Erledigung braucht, mobil verfügbar.

Insgesamt werden Brüche bei der Übertragung vermieden. Die Daten, wie Arbeitslisteneinträge und Untersuchungsergebnisse, bleiben im gleichen System. Verteilte Listen könnten zudem Kollaboration erleichtern.

2.4 Tablets

Im Allgemeinen beschreibt der Begriff „Tablet“ ein auf den Bildschirm reduziertes digitales Endgerät (siehe Abbildung 2.5). Es wird durch direkte Manipulation, also Fingerberührungen oder mithilfe eines Stylus, bedient.



(a) Dynabook, 1972 [30]



(b) Toshiba Portege M200 [35]



(c) Apple iPad, 2010

Abbildung 2.5: Beispiele für Tablet Computer

Tablet-Computer sind von ihrem Ursprung her nicht prinzipiell auf den Formfaktor und die speziellen Eigenschaften der heutigen, unter dem Begriff „Tablet“ bekannten Geräte beschränkt. Man kann aber sagen, dass die spezielle Wahl der Produkteigenschaften erst die Popularität und den Gedanken des „Post-PC“ – also des Geräts, welches die Nachfolge des PCs antritt – gangbar gemacht haben.

Die Geschichte von Tablet-Computern beginnt 1972 im Xerox Parc mit dem DynaBook [30] (siehe Abbildung 2.5a auf der vorherigen Seite). Dieses Konzept hat zwar eine physische Tastatur, ist aber vom Formfaktor und der Benutzung (inklusive e-Books) ähnlich.

Nach Weiser [60] sind Tablets ein Teil des Konzept des *ubiquitous computing*, bei dem Computer allgegenwärtig sind und in den Hintergrund treten:

„Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives.“



Abbildung 2.6: Vision des ubiquitären Computers. Zu sehen sind im Hintergrund ein *board*, in den Händen der zwei Personen im Vordergrund *pads* und auf dem Tisch mehrere *pads* (weiße, kleine Quadrate). Quelle: PARC

Dabei sollten Tablets nur ein Formfaktor unter mehreren sein: Verschiedene Geräte von Post-It- (*tabs*) über Magazin- (*pads*) bis Pinnwandgröße (*boards*) sollen jeden Einsatzzweck abdecken (siehe Abbildung 2.6). Es scheint so, als ob sich diese Vision bestätigt. Smartphones können als erster Schritt in diese Richtung gedeutet werden [8]. Tablets, die sich mit ihrer Magazin-Ähnlichkeit zum Lesen und zum Betrachten von Multimedia-Inhalten anbieten, sind somit der nächste Schritt. Durch ihre Größe eignen sie sich auch besser zur Texteingabe.

Noch in den fünfziger Jahren waren Computer, wie ENIAC¹, raumgroße industrielle Maschinen. Durch die Anbindung an Oszilloskope als Ausgabegeräte und Tastaturen anstelle von Lochkarten wurden sie nach und nach auch für Laien zugänglicher. Durch die Vernetzung mehrerer Rechner wurden schließlich die Anwendungen möglich, die wir heute täglich nutzen. Während bis vor etwa zehn bis zwanzig Jahren Fortschritte im Computer durch Entwicklung und das Hinzu-

nehmen von Komponenten gemacht wurden, ist es heutzutage eher so, dass immer mehr Komponenten ausgeblendet werden (siehe Abbildung 2.7 auf der nächsten Seite). Notebooks lassen durch Wireless Local Area Network (WLAN) das Netzwerk unsichtbar werden und bei Smartphones und Tablets entfällt

der wahrnehmbare Teil des Computers, womit nur der Bildschirm und damit die direkte Manipulation als Benutzerschnittstelle bleiben.

In der Klinik sind Tablets gut geeignet, um Eingaben zu tätigen und beispielsweise Röntgenbilder zu betrachten. Durch die Größe können auch mehrere Benutzer

¹ENIAC: Electronic Numerical Integrator and Computer, 1946 von der US-Armee eingesetzter Rechner um ballistische Tabellen zu berechnen.

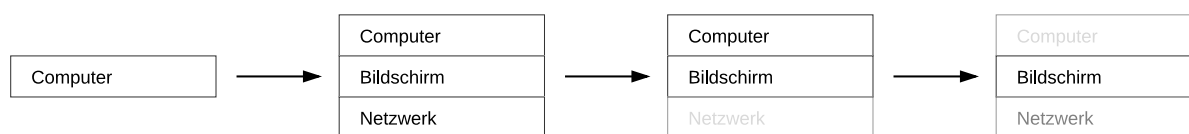


Abbildung 2.7: Wahrnehmbare Teile des Computers und deren Entwicklung in Richtung von Tablets [37]

zusammen Inhalte betrachten und diese gemeinsam diskutieren. So werden Tablets zum Beispiel an der Berliner Charité eingesetzt, um Einblick in die virtuelle Patientenakten von Krebspatienten zu geben [18, S. 2]. Über spezielle Suchfilter lassen sich schnell Patienten mit ähnlichen Diagnosen finden, um so die Therapie zu verbessern.

Es wäre auch ein integriertes Diktiergerät denkbar, so dass die Ärzte ihre Notizen sofort am Krankenbett machen können, ohne sich die Informationen bis zum Dienstzimmer entweder merken oder aufschreiben zu müssen. Genauso wäre es sinnvoll für die Dokumentation, mit dem Gerät Fotos von Wunden oder Operationsnarben des Patienten machen zu können. Weiterhin wäre die Identifikation der Patienten mittels Barcode am Bett möglich, so dass keine Verwechslungen vorkommen können.

Dadurch, dass die Einträge digital vorliegen, können sie archiviert werden. So haben Ärzt/innen/e jederzeit die Möglichkeit, in der Vergangenheit liegende Untersuchungen und Therapiemaßnahmen zurück ins Gedächtnis zu holen. Hier helfen auch Sprachmemos und Aufnahmen mit der integrierten Kamera.

Damit ein Tablet in der Klinik auf Inhalte zugreifen kann, ist es essentiell, dass die Geräte Zugriff auf die im Krankenhaus eingesetzten Datenmanagementsysteme haben (siehe Abschnitt 2.2 auf Seite 7).

Um die Daten zwischen Gerät und System abzugleichen, kann entweder eine Offline- oder eine Online-Synchronisierung eingesetzt werden. Bei der Offline-Synchronisierung muss das Tablet zum Synchronisieren an einen Host-Rechner angeschlossen werden, hat also nicht immer die aktuellen Informationen. Für die Online-Synchronisierung wird ein aktives Funknetzwerk benötigt. Durch dieses erhält das Tablet immer die aktuellsten Daten. WLAN ist noch kein Standard im Krankenhaus sowohl aus Gründen der Gerätesicherheit, die je nach Station zu klären ist, als auch der Sicherheit der Patientendaten.

Visite unter Prozessaspekten

Um konkrete Daten über Visiten im Krankenhaus zu erhalten, wurden vier Visiten auf vier verschiedenen Stationen beobachtet. Diese sind:

- Eine Visite in der Notaufnahme, die ohne Rechnerunterstützung auskommt,
- eine in der inneren Medizin, die aufgeteilt ist in Bearbeitung am Rechner und der Visite am Patientenbett,
- eine weitere in der Orthopädie, die mit mobiler Tabletunterstützung durchgeführt wird und
- eine Chefarztvisite auf der Chirurgie.

Das Ziel der Beobachtung war es, die Visiten hinsichtlich der Prozesse zu untersuchen. Zu diesem Zweck beobachtete und protokollierte ich die Visiten passiv. Dabei wurden der Ablauf und Zeiten mitgeschrieben. Dazu nutzte ich das Formular in Anhang A auf Seite 99. Für jeden Patienten sollte eines dieser Formulare verwendet werden. Es wurden die verwendeten und benötigten Daten, die beteiligten Personen, sowie ein kurzer Gesamteindruck mitgeschrieben. Allgemeine Informationen zur Station sollten im Formular auf Seite 98 notiert werden. Bei der Konzeption dieser beiden Formulare wurde von der klassischen, in vier Phasen eingeteilte Struktur der Visite ausgegangen. Während der Beobachtung stellte sich jedoch heraus, dass diese Struktur wenn überhaupt, dann nur sehr lose angewandt werden kann. Aus diesem Grund wurde auf die Verwendung der Formulare verzichtet und stattdessen frei protokolliert.

In den folgenden Abschnitten werden die beobachteten Visiten vorgestellt und auf ihre jeweiligen Besonderheiten eingegangen. Dabei werden die Stationen charakterisiert, die Visite beschrieben, um dann schließlich Probleme aufzuführen, die sich daraus ergeben. Zusätzlich wurden zu allen Stationen Prozessdiagramme erstellt, die den Ablauf der Visite skizzieren. Die Prozesse in den folgenden Abschnitten sind in BPMN 2.0 modelliert. Sie sollen keine ausführbaren Prozesse darstellen, sondern dienen der Vergleichbarkeit der Prozesse.

Tabelle 3.1 auf Seite 19 zeigt einen Überblick über die betrachteten Stationen. Die Liegezeiten der Patienten, die Aufgabe der Visite und teilnehmende Rollen sind als Hauptunterschiede zu erkennen.

Die Visite in der Orthopädie im Universitäts- und Rehabilitationskliniken Ulm (RKU) ist als Besonderheit herauszuheben, denn hier wird bereits ein Tablet-Computer eingesetzt, um Patientendaten mobil am Bett verfügbar zu haben.

3.1 Innere Medizin

Station

Die betrachtete Station in der inneren Medizin im Universitätsklinikum Ulm behandelt hauptsächlich Diabetespatienten. Aus diesem Grund sind Laborbefunde für die Visite von besonderer Bedeutung. Nur in seltenen Fällen werden Röntgenbilder gebraucht.

Wie in der gesamten Universitätsklinik wird auf der inneren Medizin ein angepasstes SAP-System zur Patientenverwaltung genutzt. Es kann von mehreren Computern auf der Station darauf zugegriffen werden. Die Patientenakte wird hauptsächlich digital verwaltet, inklusive Entlassungsbriefen, Anforderungen und Untersuchungsergebnissen. Nicht digital vorliegende Teile der Patientenakte werden nach der Entlassung des Patienten eingescannt und in das System eingespeist.

Visite

Auf der internistischen Station ist die Visite zweigeteilt: Im Dienstzimmer werden am Computer die Patienten besprochen. Anschließend teilen sich die Ärzte auf und besuchen die Patienten in ihren Zimmern (siehe Abbildung 3.1 auf Seite 20).

Da der Patient während diesem ersten Teil nicht anwesend ist, wird dies oft *Kurvenvisite* genannt. Die Besprechung am Computer findet zwischen den beiden Stationsärzten und zwei Pflegekräften statt. Die Meinungen und Beobachtungen der Pflegekräfte sind dabei wichtig, da keine Möglichkeit besteht, den Patienten zu fragen. Etwa 10–20% der Aussagen sind vom Pflegepersonal. Des Weiteren werden die Daten aus der Pflegedokumentation, sowie Befunde und Untersuchungsergebnisse aus dem KIS herangezogen.

Da diese Gespräche im Dienstzimmer der Ärzte stattfinden, nutzen die Ärztinnen und Ärzte oft die Möglichkeit, Telefongespräche zu führen. Diese sind zum Beispiel nötig, um andere Krankenhäuser nach der dortigen Medikationen eines Patienten zu fragen, die Meinung einer Chefärztin/eines Chefarztes einzuholen oder Termine zu vereinbaren oder zu verschieben. Änderungen am Entlassungsbrief aufgrund von Befunden werden gleich im System vorgenommen. Medikationsänderungen werden direkt abgezeichnet. Da der Patient nicht anwesend ist, können Ärzte und Pflegepersonal sehr viel offener über den Zustand des Patienten diskutieren.

Aufgaben, die nicht sofort erledigt werden können, notieren sich die Ärztinnen und Ärzte auf persönlichen Laufzetteln. Zusätzlich wird auf dem Computer ein Dokument geführt, welches in Tabellenform den Zustand der Station abbildet. Zu jedem belegtem Bett werden hier Patientennamen, Diagnose, anstehende Untersuchun-

Tabelle 3.1: Allgemeine Informationen über die betrachteten Stationen

	Innere Medizin	Notaufnahme	Orthopädie (Querschnitt)	Chirurgie (Chefvisite)
Krankenhaus	Universitätsklinikum Ulm	Universitätsklinikum Ulm	RKU	Universitätsklinikum Ulm
Betten	25	12	35	>100
Liegezeiten	25 Tage, Wochen	12 Stunden	35 Wochen, Monate	wenige Tage bis zu einem Monat
Häufigkeit der Visite	Zweimal täglich	Dreimal täglich	Einmal täglich	Einmal in der Woche
Aufgabe der Visite	Tägliches Bild	Informationsweitergabe zwischen den Schichten	Tägliches Bild	Übersicht und Kontrolle für den Chefarzt
Beteiligte Rollen	2 Stationsärzte, 1–2 Pflegekräfte	6–8 Stationsärzte, Oberarzt und eventuell weitere Spezialisten	3 Stationsärzte, 1–2 Pflegekräfte, Sport-, Ergo- und Physiotherapeut	Chefarzt, Vertretung sowie Ärzte der jeweiligen besuchten Station
Zeit pro Patient KIS	7 + 3,5 Min ERP, SAP	4 Minuten ERP, SAP	4 Minuten MCC, Meierhofer AG	3 Minuten ERP, SAP
Daten, mobil verfügbar				
Vitaldaten	✓(Pflegedokumentation)	✓(Pflegedokumentation)	✓(Pflegedokumentation)	✓(Pflegedokumentation)
Medikation	✓(Pflegedokumentation)	✓(Pflegedokumentation)	✓(Pflegedokumentation)	✓(Pflegedokumentation)
Bildgebende Diagnostik	✗	✗	✓(Tablet)	✓(ausgedruckt)
Laborbefunde	✗	✗	✓(Tablet)	✓(ausgedruckt)

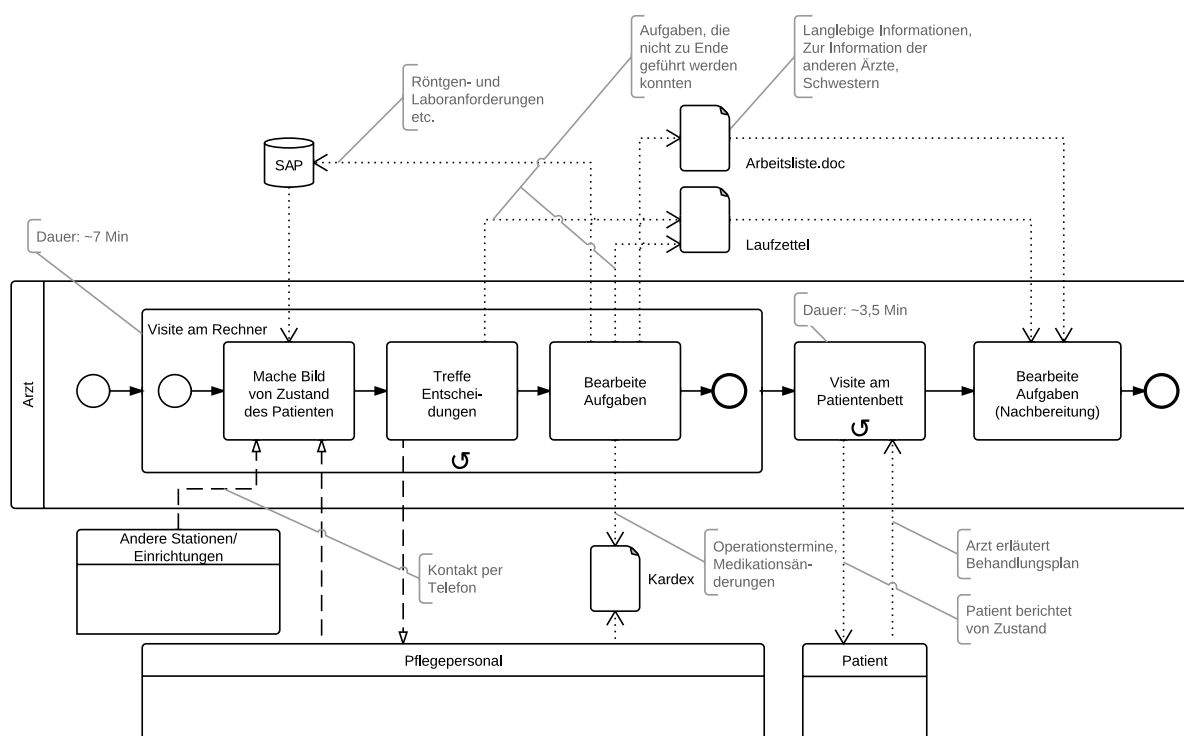


Abbildung 3.1: Prozessmodell Visite in der Inneren Medizin

gen sowie das geplante Entlassungsdatum festgehalten. Je nach Präferenz der Ärzt/innen/e füllen entweder der Arzt/die Ärztin selber oder der/die Stationssekretär/in die Anforderung aus.

Im Anschluss besuchen die Ärzt/innen/e die Patienten am Bett. Um Zeit zu sparen, übernimmt jeder der beiden Stationsarzt/innen/e eine Hälfte der Station. Die Patienten werden nach ihrem Befinden gefragt und ihnen der Behandlungsplan erklärt. Dabei werden keine Notizen geführt oder mitgenommen.

Probleme

Dadurch, dass die meisten Aufgaben sofort erledigt werden können, wird Zeit gespart. Fehler und Fehlentscheidungen aufgrund von Zeitabständen zwischen Aufgabenerfassung und -bearbeitung werden so unwahrscheinlicher. Allerdings kann die Meinung der Patienten nicht genutzt werden. Diese Informationen können nur indirekt durch die Beobachtungen des Pflegepersonals einfließen.

Da der Patient erst später besucht wird, kann es zu Diskrepanzen zwischen den Entscheidungen im Dienstzimmer und der Entscheidung, die die Ärzt/innen/e dem Patienten mitteilen, kommen. Wird die neue Entscheidung nicht nachträglich dokumentiert, handeln die Pflegekräfte möglicherweise noch nach der alten Anweisung.

Ein weiteres Problem ist, dass zwei verschiedene Systeme genutzt werden, um den Zustand der Station abzubilden: Der Laufzettel der einzelnen Ärzt/innen/e und das gemeinsam geführte Dokument am Computer.

Die Zerteilung der Visite zeigt, dass die Digitalisierung von Teilen der Patientendaten Nachteile für die Patientenbetreuung hat. Es musste eine Entscheidung getroffen werden zwischen einer Visite am Patientenbett ohne Patientendaten und einer Visite am Computer ohne Möglichkeit, den Patienten zu untersuchen. Dies macht deutlich, dass die mobile Verfügbarkeit von Patientendaten wichtig ist, um eine patientenzentrierte Visite zu ermöglichen. Die Befragung von Ärzten hat ergeben, dass die meisten eine Visite am Patientenbett vorziehen würden.

3.2 Notaufnahme

Station

Kurze Stationsaufenthalte und Verlegungen innerhalb einiger Stunden auf andere Stationen sind charakteristisch für die Notaufnahme. Die Hauptaufgabe dieser Station ist die schnelle Erstellung einer Diagnose. Aufgrund dieser werden Maßnahmen ergriffen und entschieden, ob und wann der Patient auf welche Station verlegt werden soll.

Ein großes Whiteboard im Dienstzimmer gibt Auskunft darüber, wie die Betten belegt sind, wer der jeweils betreuende Arzt ist und welche Maßnahmen anstehen. Die Notaufnahme nutzt wie die gesamte Universitätsklinik SAP zur Patientenverwaltung. Siehe dazu Abschnitt 3.1 auf Seite 18.

Visite

Die Visite dient rein der Übergabe zwischen den Schichten. Sie ist entsprechend effektiv daraufhin ausgerichtet, dass der übernehmende Oberarzt und die Assistenzärzte schnell und zielgerichtet über alle Patienten informiert werden. Da beide Schichten teilnehmen, sind zwei Oberärzte, sechs bis acht Assistenzärzte, sowie bis zu sechs Studierende anwesend.

Vor der Visite bereiten die Ärzte, deren Schicht endet, verdichtete Berichte zu ihren Patienten vor (siehe Abbildung 3.2 auf der nächsten Seite). Während der Visite kann so jeder Patient den Ärzten der nächsten Schicht vorgestellt werden. Mithilfe dieses Berichts kann der Oberarzt schnell einen umfassenden Eindruck gewinnen. Anweisungen und Patienteninformationen werden von allen Ärzten mitgeschrieben, deren Schicht gerade beginnt, da erst nach der Visite entschieden wird, wem welcher Patient zugeteilt wird. Nach der Visite werden die Aufgaben auf das Patientenwhiteboard notiert. So können die Schichtleiter/innen einen Überblick über die Abläufe und ausstehende Aufgaben der Station bekommen.

Während der Visite sind nur die Vitaldaten der Patienten schriftlich am Bett verfügbar. Die Berichte der Ärzt/innen/e sind allerdings so ausgiebig, dass es nie zu fehlenden Informationen kommt.

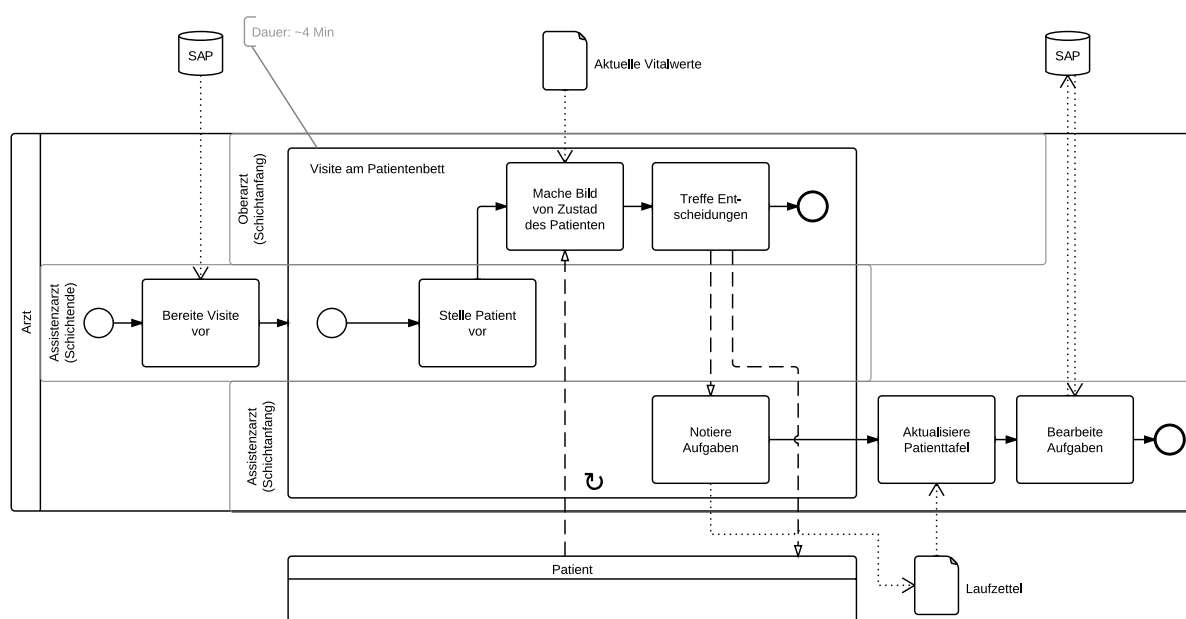


Abbildung 3.2: Prozessmodell Visite in der Notaufnahme

Probleme

Das Hauptproblem in der Notaufnahme sind wie auf den anderen Stationen ständige Ablenkungen. Verstärkt durch die große Anzahl an Teilnehmern, wird fast jede Visite durch Telefonanrufe, Notfälle oder andere Unterbrechungen gestört. Diese Ablenkungen können potenziell dazu führen, dass Aufgaben vergessen werden.

3.3 Orthopädie

Station

Auf der orthopädischen Station der RKU werden hauptsächlich querschnittsgelähmte Patienten behandelt. Die Liegezeiten der Patienten betragen meist mehrere Monate. Bei der Visite sind neben Ärzten und Pflegepersonal mehrere Therapeuten anwesend: Ergo-, Physio- und Sporttherapeuten betreuen die Patienten in der Regel einmal am Tag und haben einen guten Einblick in den Genesungsverlauf des Patienten. Da die Patienten lange auf der Stationen bleiben, wird das Verhältnis zu den Mitarbeitern intensiver. Gleichzeitig sind sie durch die Art ihrer Krankheit über ihren Zustand besser informiert als andere Patienten, da die Erkrankung weitreichende Folgen für ihr weiteres Leben hat.

Auf der Station wird das KIS Meierhofer Clinical Competence (MCC) eingesetzt. Dieses läuft auf einem Citrix-Server¹ und ist dem Tailoring-Ansatz

¹Citrix entwickelt Application Server Lösungen, d.h. die Anwendungen werden auf einem Server ausgeführt und auf Client-Computern angezeigt. Diese brauchen entsprechend weniger Rechenleistung (Thin Clients).

entsprechend auf die Bedürfnisse und Anforderungen des RKU angepasst. Die Ärzt/innen/e können von Desktop-Computern im Dienstzimmer auf die dort gespeicherten Daten zugreifen. Der Drucker, über den Röntgenanforderungen und andere Formulare ausgedruckt werden, steht ebenfalls in diesem Zimmer. Neben dem stationären Zugriff besteht noch die Möglichkeit, auf Visite einen Visitenwagen mitzunehmen, in den unter anderem ein Computer integriert ist. Außerdem sind auf dieser Station mit dem Mobile Clinical Assistant von Motion Computing, Inc. (siehe Abbildung 3.3) Tablet-Computer mit Stiftbedienung im Einsatz. Diese dienen als Thin-Clients für das MCC. Die Benutzerschnittstelle ist dabei die gleiche wie am Desktoprechner. Diese Mobilität wird durch ein stationsweites WLAN ermöglicht.



Abbildung 3.3: Mobile Clinical Assistant von Motion Computing, Inc.

Visite

Von der Möglichkeit, bei der Visite ein Tablet mitzunehmen, machen nicht alle Ärzt/innen/e Gebrauch. Während der Visite am Patientenbett können sie mit dem Tablet Laborbefunde einsehen, Röntgenbilder abrufen und verschiedene Untersuchungen anordnen. Je nach Untersuchung muss die Anordnung nach der Visite unterschrieben werden.

Ansonsten wird die Visite klassisch durchgeführt, das heißt die Ärztin/der Arzt übernimmt die Gesprächsführung; Pflegekräfte und Therapeuten berichten gegebenenfalls zusätzlich (siehe Abbildung 3.4 auf der nächsten Seite). Die Patienten tragen, verglichen mit anderen Stationen, relativ viel zur Visite bei. Sie berichten über Körperfunktionen, ihre Fortschritte und Probleme. Auf dem Tablet werden teilweise Röntgenbilder und Laborbefunde eingesehen.

Sehr oft wird die Möglichkeit genutzt, aus dem Programm heraus Röntgenanforderungen zu erstellen. Dabei wird durch einen Klick im Patienten-Fenster eine Word-Vorlage geöffnet, die bereits mit den Patientendaten vorgefüllt ist. Der Benutzer muss nun noch die Art der Untersuchung, bisherige Befunde und den Grund der Untersuchung angeben. Diese Daten müssen über eine kleine On-Screen-Tastatur eingegeben werden (siehe Abbildung 3.5). Schließlich wird das Word-Dokument gedruckt und geschlossen. Am Ende der Visite findet die Ärztin/der Arzt die Anforderung im Drucker im Dienstzimmer. Er unterschreibt sie und gibt sie in die Hauspost. Neben der Röntgenanforderung kann im Programm auch eine Laboruntersuchung des Blutes angefordert werden. Im Dienstzimmer steht hierzu ein Drucker, welcher Aufkleber für die Proben druckt. Die Probe muss

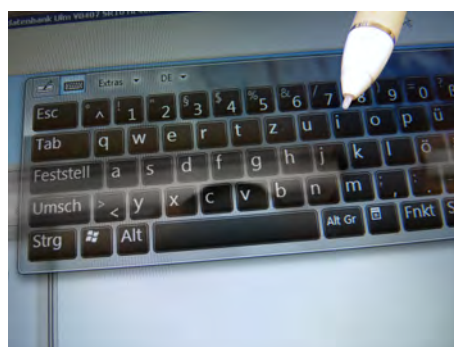


Abbildung 3.5: Bildschirmtastatur und Stift des Mobile Clinical Assistant

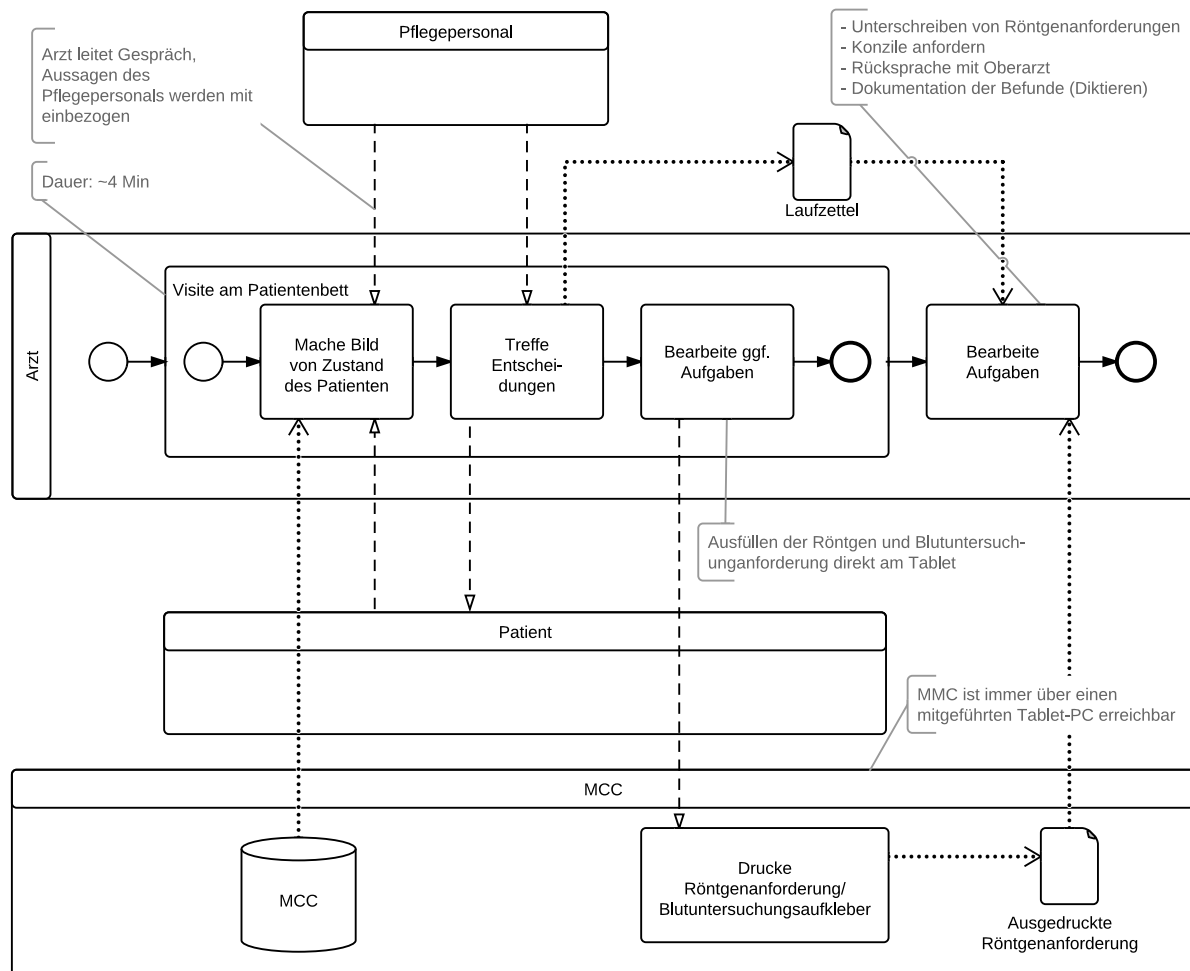


Abbildung 3.4: Prozessmodell Visite in der Orthopädie

dann nur noch genommen, mit dem Aufkleber versehen und in die Hauspost gegeben werden.

Probleme

Die Software auf dem Tablet ist nicht für mobile Anwendungen angepasst, weshalb manche Interaktionen (beispielsweise zu Konsilergebnissen navigieren oder Tasteingaben) den Fluss der Visite bremsen. Das System meldet sich schon nach kurzer Zeit aus Sicherheitsgründen ab. Deshalb muss der Benutzer sich sehr oft anmelden oder daran denken, ab und zu den Bildschirm zu berühren.

Problematisch ist die Verteilung der Daten in mehreren Systemen. Neben dem MCC werden weitere Systeme für Röntgenbilder und Laborbefunde betrieben, bei denen sich der Benutzer ebenso anmelden muss.

Durch das Gerät muss der Arzt während der Visite sehr viele Dinge gleichzeitig tun: Tablet mit Stift halten, ein Diktiergerät benutzen, den Patienten untersuchen und mit ihm kommunizieren.

3.4 Chirurgie

Station

Es wurde in dieser Betrachtung eine Chefvisite an einer Klinik für Unfall-, Hand-, Plastischer- und Wiederherstellungschirurgie analysiert. Mit über 100 Betten handelt es sich um eine große Klinik, die in weitere Stationen eingeteilt ist. Sie ist Teil des Universitätsklinikums und setzt das gleiche SAP-System ein.

Visite

Die betrachtete Visite auf der Chirurgie ist insofern von den anderen beschriebenen Visiten zu unterscheiden, als dass es sich hier um eine Chefvisite handelt. Diese findet einmal in der Woche statt. Dabei wird nicht nur eine Station besucht, sondern die Chefärzti/der Chefarzt besucht alle ihm unterstehenden Stationen nacheinander. An der Visite nehmen die Chefärztin/der Chefarzt, seine Vertretung, ein bis zwei Assistenzärzt/innen/e und, je nach Station, Pflegekräfte und Physiotherapeuten teil. Beim Übergang zur nächsten Station wechseln die Assistenzärzt/innen/e und die Pflegekräfte.

Die Dynamik bei der Chefvisite ist hoch, da viele Patienten besucht werden müssen. Pro Patient bleiben nur wenige Minuten, in denen die Chefärzt/innen/e meist die Röntgenbilder routiniert betrachten und, je nach Zustand des Patienten, Anweisungen für weitere Untersuchungen geben.

Probleme

Das Hauptproblem der Visite in der Chirurgie ist, dass die Visite gut vorbereitet werden muss. Die Ärzt/innen/e suchen vor der Visite relevante Röntgenbilder heraus, um sie schnell dem Chefarzt vorstellen zu können. Diese Röntgenbilder werden aus-

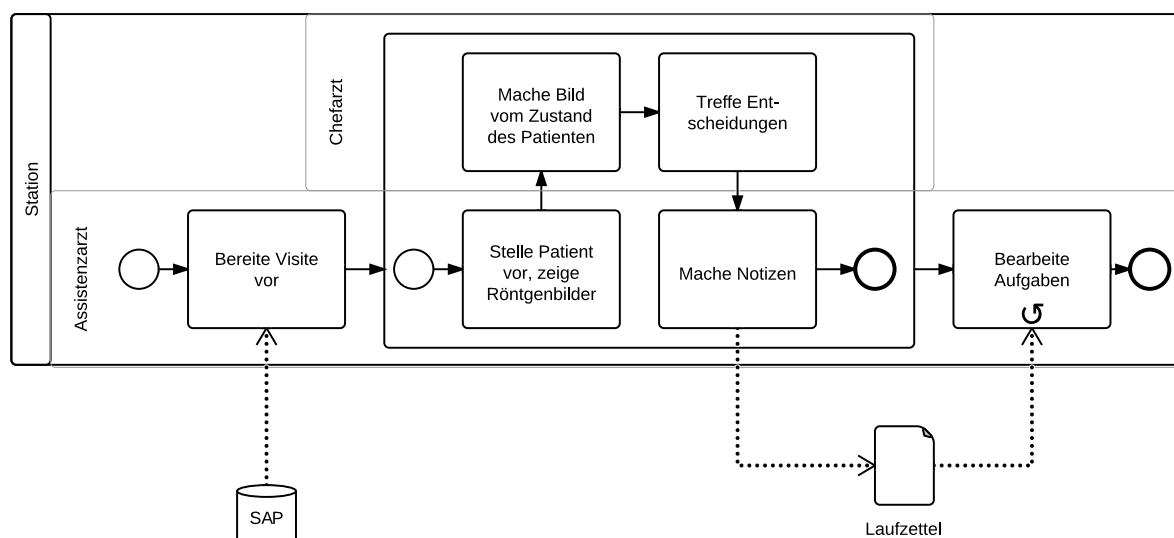


Abbildung 3.6: Prozessmodell Visite in der Chirurgie

gedruckt und müssen dann während der Visite mitgeführt werden.

Bei der großen Anzahl von Informationen bleibt nicht viel Zeit für den einzelnen Patienten. Dafür ist diese Visite im Gegensatz zu den anderen Visiten aber auch nicht ausgelegt.

3.5 Zusammenfassung

Bei der Untersuchung hat sich gezeigt, dass jede der beobachteten Visiten anders organisiert ist. Diese Unterschiede und die Dynamik innerhalb der einzelnen Visiten lassen den Schluss zu, dass sich diese Prozesse nicht in einen großen Workflow integrieren lassen, der die gesamte Visite umspannt. Stattdessen können bei allen Visiten um den Hauptprozess „Visite“ weitere Unterprozesse erkannt werden.

Aufgefallen sind außerdem der fehlende Patientenkontakt während der Besprechung auf der inneren Medizin und auf allen Stationen die Nutzung eines Notizzettels, um anstehende Arbeitsschritte zu notieren.

Zu diesen erwähnten „Peripherie“-Workflows gehört zum Beispiel die Röntgenuntersuchung. Bei diesem Prozess sind neben den initiiierenden Arzt/inn/en noch die beiden Rollen Pflegepersonal und Röntgenstation beteiligt (siehe Abbildung 3.7 auf der nächsten Seite). Die Ärztin/der Arzt leitet die Anforderung mit Anamnese (Beispiel: Hüftprothese), Fragestellung (Beispiel: Lockerung der Hüfte) und Untersuchungsziel (Beispiel: Hüfte links, zweischichtig) an die Röntgenstation weiter. Von der Art der Anforderung hängt ab, ob der Patient sobald wie möglich abgeholt wird, oder ob ein Termin gewünscht ist. Ist letzteres gewünscht, so kann das Pflegepersonal beispielsweise vor der Untersuchung Kontrastmittel verabreichen. Nach der Untersuchung schickt das Labor die Bilder elektronisch und per Hauspost an die Station,

wo die behandelnden Ärzt/innen/e die Ergebnisse analysieren und den Befund dokumentieren.

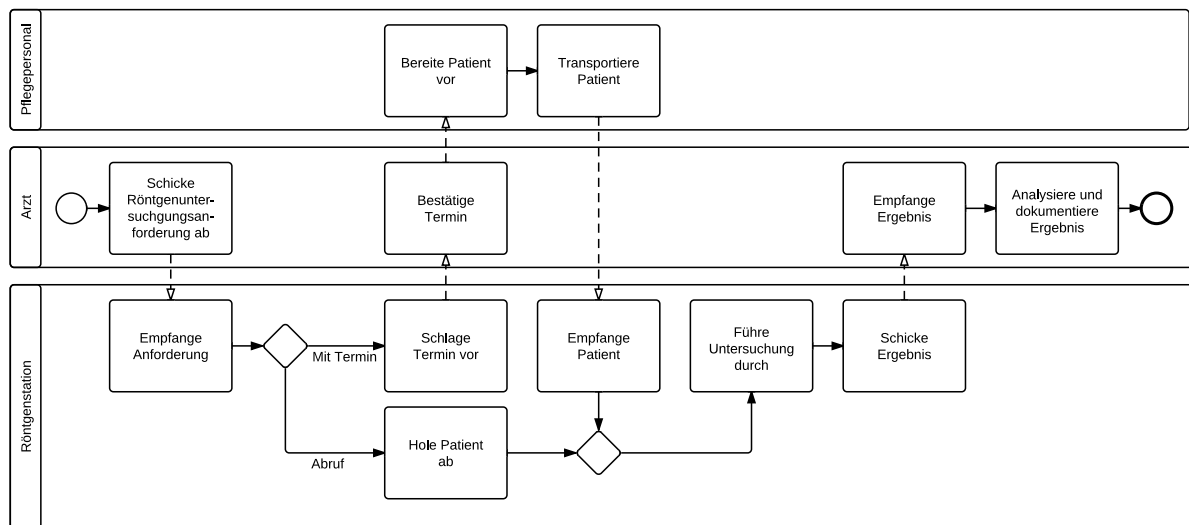


Abbildung 3.7: Prozessmodell Röntgenuntersuchung

Als zweites Beispiel sei an dieser Stelle die Blutuntersuchung erwähnt (siehe Abbildung 3.8). Andere Laboruntersuchungen laufen ähnlich ab. An die Position der Röntgenstation tritt hier das Labor. Nachdem das Pflegepersonal über die Blutuntersuchung informiert ist, trifft es Vorbereitungen, beispielsweise wenn der Patient für die Untersuchung nüchtern sein muss. Außerdem wird das Röhrchen vorbereitet, indem es mit einem Aufkleber versehen wird, der die Probe eindeutig dem Patienten zuordnet. Nun nimmt eine Ärztin/ein Arzt dem Patienten Blut ab, woraufhin das Pflegepersonal die Blutprobe verschickt. Das Ergebnis erhält die Ärztin/der Arzt wieder sowohl auf Papier als auch elektronisch. Er analysiert es, dokumentiert den Befund und zeichnet es ab.

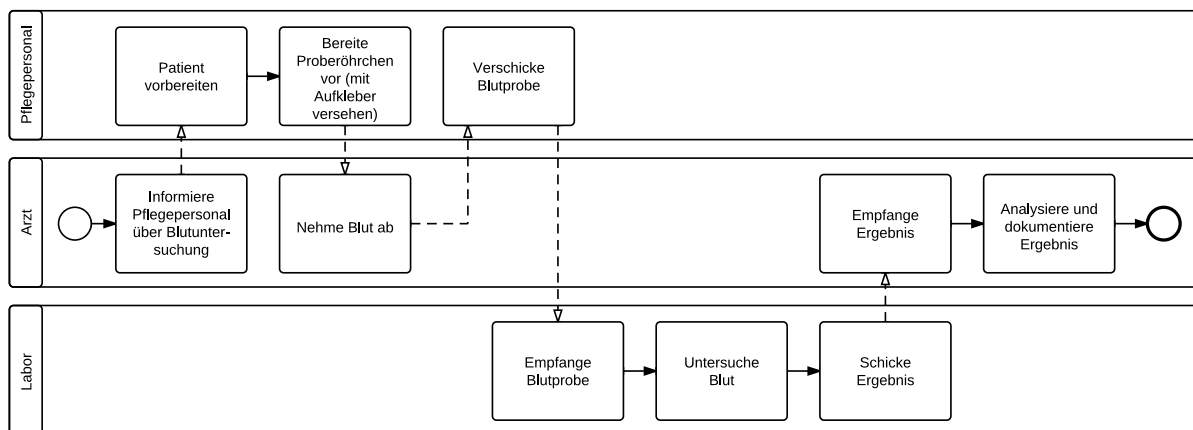


Abbildung 3.8: Prozessmodell Anforderung einer Blutuntersuchung

Diese Prozesse sind Beispiele dafür, wie Abläufe in der Peripherie der Visite prozesstechnisch, also beispielsweise durch ein WfMS unterstützt werden können. So könnte Ärztinnen und Ärzte jederzeit den Status der von ihnen angeordneten Untersuchungen überprüfen.

Es ist nun aber nicht zielführend diese Prozesse nur an einem Desktoprechner den Ärzt/inn/en zugänglich zu machen. Stattdessen sollten die Benutzer mobil unterstützt werden, damit sie auch während der Visite auf die Prozesse und ihre Aufgaben zugreifen können. Die Entwicklung eines solchen Systems auf Basis eines Tablet-Computers wird im folgenden Kapitel geschildert.

4

Anforderungsanalyse

Nach den Betrachtungen im vorigen Kapitel sind zusammengefasst vier Probleme der Visite identifizierbar:

- Nicht definierte Prozesse,
- Informationsverlust durch Medienbrüche,
- keine Nachverfolgbarkeit der erledigten Aufgaben und
- fehlende Mobilität der Daten.

Diese soll *MedicalDo*, abgekürzt *MEDo*, lösen. Dazu wird Aufgabenmanagement, Prozessmanagement und die Anzeige von Patientendaten in einer Applikation vereint. Diese soll auf einem iPad installiert werden, so dass alle diese Features mobil verfügbar sind. Das System soll als Prototyp auf Einsetzbarkeit auf drei der im vorangegangenen Kapitel vorgestellten Stationen geprüft werden.

Durch die Unterstützung von Ärztinnen und Ärzten bei der Bearbeitung von Aufgaben können Medienbrüche und Fehlinformationen verhindert werden. Aufgaben, die einen Prozess auslösen, sollten durch ein WfMS unterstützt werden. Dieses blendet automatisch weitere Anweisungen in Arbeitslisten der beteiligten Rollen ein. So wird eine bessere Kooperation und Koordination der Arbeit ermöglicht. Als logischer Schritt sollten auf einem mobilen Endgerät Patienteninformationen eingesehen werden können, wie es bereits auf der Orthopädie mit dem Tablet PC möglich ist.

In diesem und den folgenden drei Kapiteln werden alle Phasen der Umsetzung sowie der Evaluation beschrieben. Zunächst werden in diesem Kapitel Anforderungen an das System zusammengetragen. Danach steht die Entwurfsphase, in der Ideen zur Architektur, Navigation, zum Layout, Framework und Styleguide gesammelt und spezifiziert werden. Die Implementationsphase umfasst die Entwicklung und Programmierung der finalen Architektur. Die Evaluierung schließt die Entwicklung des Prototypen ab. Abbildung 4.1 auf der nächsten Seite stellt die Zusammenhänge dieser Kapitel grafisch dar.

Die Erfassung der Anforderungen ist unterteilt in solche, die an ein visitenunterstützendes System gestellt werden und solche, die sich durch die Wahl des Mediums

Tablet ergeben. Abschliessend werden Szenarien vorgestellt, die mit dem System unterstützt werden sollen.

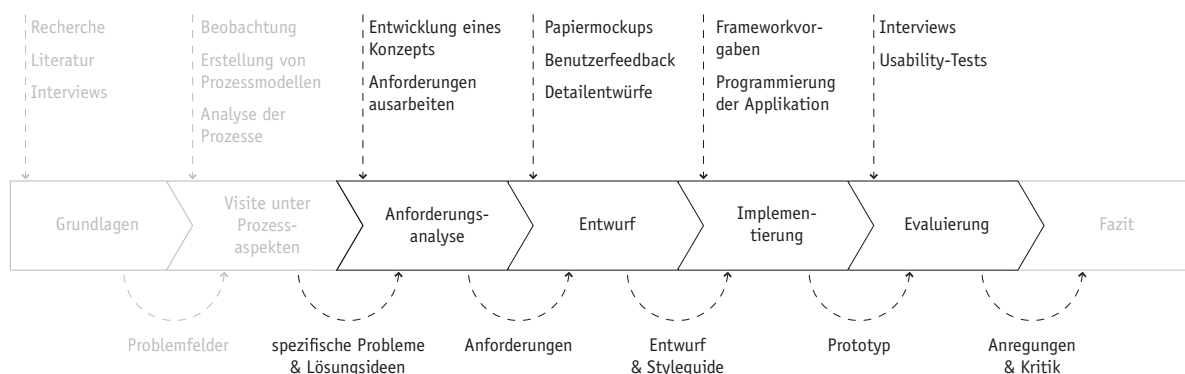


Abbildung 4.1: Übersicht über die Phasen bis zur Evaluierung des Prototyps

4.1 Anforderungen an den Laufzettel

Basierend auf der Beobachtung im Kapitel „Visite unter Prozessaspekten“ bestehen folgende Anforderungen an ein zu implementierendes System:

Abbildung eines papierbasierten Laufzettels Die Hauptaufgabe eines Laufzettels ist das Auflisten und Hinzufügen einfacher Texteinträge. Neben dem Text des Eintrages wird der Patientennamen festgehalten. Dabei soll der Aufgabentext als Freitext jederzeit erhalten bleiben und als primäre Eingabe anderen Eingabemethoden vorgezogen werden.

Einbeziehung von zusätzlichen Daten Es können neben den bereits erwähnten Daten weitere Informationen für Arbeitslisteneinträge protokolliert werden. Zusammengefasst ergeben sich folgende Daten:

Patientenname

Aktivität Text des Aufgabe

Zeitpunkt aufgenommen Erstellungsdatum der Aufgabe.

Zeitpunkt bearbeitet Datum, an dem die Aufgabe als erledigt markiert wurde.

Benutzer aufgenommen Name des Anwenders, der die Aufgabe erstellt hat.

Benutzer bearbeitet Name des Benutzers, der die Aufgabe abgeschlossen hat.

zugeordnete Daten Beispielsweise Untersuchungsergebnisse und Befunde.

Dabei können Zeiten sowie Benutzerdaten automatisch vom System protokolliert werden. Vom Benutzer müssen also der Patient und der Aufgabentext explizit eingetragen werden. Letzterer benötigt bei der Eingabe mehr Zeit und muss dementsprechend vom System in unterschiedlicher Art unterstützt werden. Siehe dazu die Anforderung „Geschwindigkeit der Eingabe.“

Arbeitslistenfunktion Der elektronische Laufzettel sollte, zusätzlich zu den oben bereits genannten Texteinträgen, auch automatische, durch Prozesse definierte Aktivitäten unterstützen.

Ein Beispiel wäre, dass ein Arzt, der eine Aufgabe „Röntgenanforderung“ erstellt hat, diesen Eintrag nach der Visite bearbeitet. Nach dem Ausfüllen und Abschicken der Röntgenanforderung wird der Eintrag aus der Liste entfernt. Durch das Abschicken wird der Prozess „Röntgenuntersuchung anfordern“ ausgeführt (siehe hierzu Abbildungen 3.7 bis 3.8 auf Seite 27).

Die jeweiligen Prozesse sollten auf die Stationen angepasst werden, entsprechend den dort anzutreffenden Anforderungen. Ein im Hintergrund eingesetztes WfMS würde eine solche Anpassbarkeit bieten.

Unterbrechbarkeit Dadurch, dass Visiten sehr unterschiedlich durchgeführt werden, muss das System eine Vielzahl von Verwendungen unterstützen. Beispielsweise sollte ein Benutzer jederzeit das Ausfüllen einer Röntgenanforderung abbrechen können, um schnell zu einem anderen Patienten zu wechseln und dort Daten einzusehen. Es sollte in einem solchen Fall automatisch ein Arbeitslisten-Eintrag erstellt werden und eine Sicherung der Daten durchgeführt werden. Zu einem späteren Zeitpunkt kann so der Benutzer die Eintragung weiterbearbeiten.

Flexibilität Nicht vorausgeplante Ausnahmen stellen ein Problem für die meisten Workflowumgebungen dar. Tritt ein solcher Fall ein, muss vom modellierten Standardablauf abgewichen werden. Die wenigsten heutigen WfMS gestatten solch einen Eingriff [45, 46].

Beim Einsatz im Krankenhaus werden viele Sonderfälle eintreten, die nicht alle im Vorfeld eingeplant werden können. So macht es Sinn, dass auch die Prozesse, die MEDo unterstützt, flexibel sind. Dafür muss eine entsprechende Benutzerschnittstelle angeboten werden.

Geschwindigkeit der Eingabe Um mit der Geschwindigkeit der Kommunikation während der Visite mithalten zu können, muss die Eingabe sehr schnell geschehen. Die Art der Texteingabe spielt dabei eine große Rolle. Denkbare Eingabemöglichkeiten wären:

- Handschrifterkennung,
- Textbausteine,
- Hardwaretastatur,
- Spracherkennung,
- virtuelle Tastatur mit Touchscreen und
- virtuelle Tastatur mit Stifteingabe.

Allerdings hat nicht nur die Texteingabe Einfluss auf die Bedienung. Das Bedienkonzept muss ebenso entsprechend ausgerichtet sein. Weitere Faktoren sind außerdem die Geschwindigkeit der Software bei der Anzeige und Bedie-

nung sowie die Geschwindigkeit der Datenspeicherung und -abfrage.

Rollen-/Mehrbenutzerunterstützung Es sollten mehrere Rollen unterstützt werden, hauptsächlich zur Unterscheidung der Rollen Arzt und Pflegepersonal. Je nach Rolle ist die Benutzerschnittstelle anzupassen. Abhängig von der Rolle sind auch die Rechte des angemeldeten Benutzers zu modellieren.

Geteilte Listen Benutzer brauchen eine Möglichkeit, ihre Arbeitslisten mit Anderen zu teilen. Denkbar wäre auch ein nur lesender Zugriff auf Listen.

Adäquate Anzeige der anstehenden Aufgaben Es sollte jederzeit für einen angemeldeten Benutzer möglich sein, alle anstehenden Aufgaben einzusehen. Er sollte so schnell wie möglich die für seine aktuelle Arbeitssituation passende Aufgabe auswählen können.

Adäquate Bearbeitung der Aufgaben Aufgaben, die ohne Handlung ausserhalb der elektronischen Umgebung bearbeitet werden können, sollten automatisch abgeschlossen werden. Sobald der Arzt das entsprechende Formular ausgefüllt hat, muss der zugeordnete Arbeitslisteneintrag als abgearbeitet angezeigt werden.

Aufgaben, die manuelle Handlungen verlangen, sollten durch das System unterstützt werden. Nach Abschluß der Arbeit sollten diese Aufgaben ebenso als erledigt markierbar sein.

Jederzeit Aufgabenbearbeitung unterbrechen Unterbrechungen sind in der Klinik an der Tagesordnung. Der Benutzer sollte somit jederzeit die Möglichkeit haben, mit einer Aufgabe zu pausieren. Auf die Pausierung hin sollte automatisch ein Arbeitslisteneintrag erstellt werden, sowie ein Zwischenstand der bisher eingegeben Daten. Nach der Unterbrechung kann die Ärztin oder der Arzt so nahtlos weiterarbeiten.

Unterstützung für Schichtübergabe Laufzettel müssen entweder immer für alle einsehbar sein oder es sollte die Möglichkeit gegeben werden, den Laufzettel an die nächste Schicht zu übergeben.

Adäquate Anzeige von Untersuchungsergebnissen Der Arzt muss sehr schnellen Zugriff auf alle bisherigen Untersuchungsergebnisse haben. Hier sind alle unnötigen Interaktionen zu vermeiden. Negativbeispiel ist hier das MCC der Meyerhofer AG: Jeder Untersuchungsbericht ist mit mindestens drei Klicks und entsprechenden Wartezeiten dazwischen verbunden (siehe Abbildung 4.2 auf der nächsten Seite).

Adäquate Benachrichtigung über Status von Untersuchungen Im Optimalfall sollten die Ärzt/innen/e automatisch über *alle* neu eingetroffenen Untersuchungsergebnisse informiert werden. Dies erfolgt idealerweise als Eintrag in der Arbeitsliste. Eine Berührung oder Klick bringt den Anwender zum Untersuchungsergebnis. Hier sollte er die Möglichkeit haben, das Untersuchungsergebnis als gelesen zu markieren und damit aus der Arbeitsliste zu entfernen.

Allerdings ist nicht nur das Untersuchungsergebnis an sich wichtig, son-

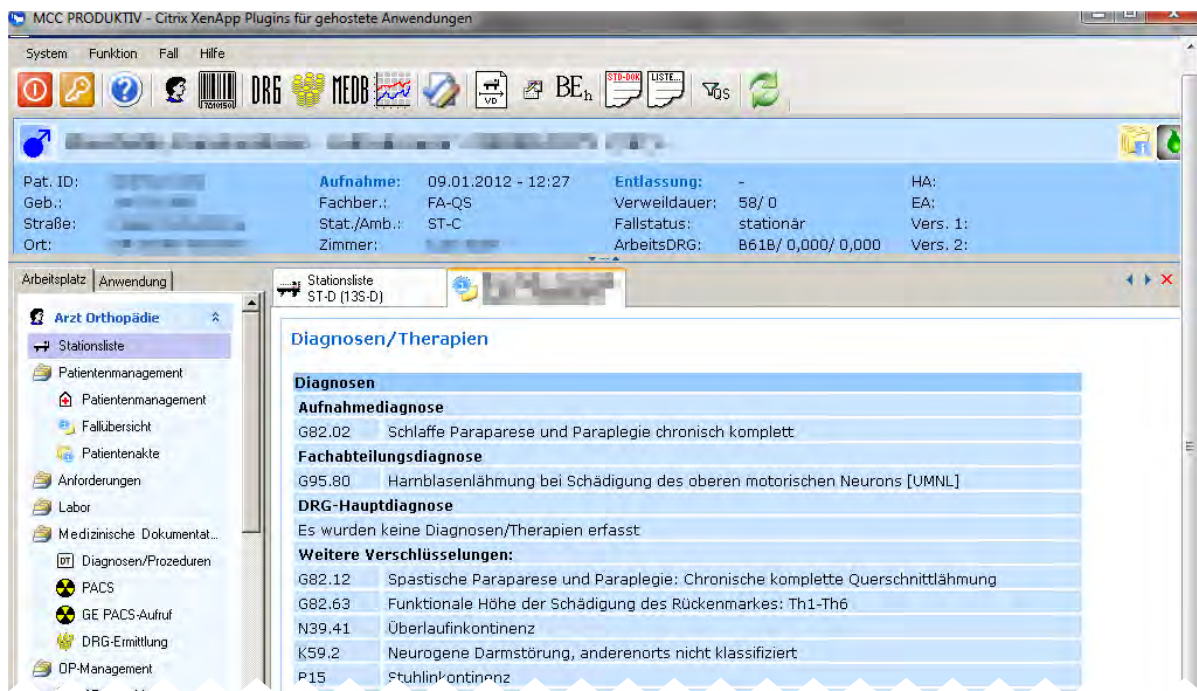


Abbildung 4.2: Patientenakte im MCC. Um zum Kumulativbefund zu gelangen, muss der Benutzer den Patienten auswählen, in der rechten Leiste „Arbeitsplatz“ anklicken und in einem Dropdown „Kumulativbefund“ auswählen.

dern auch der Status der Untersuchungsanordnung. Zu unterscheiden wären folgende Stati:

- Angeordnet,
- Termin steht fest,
- Untersuchung abgeschlossen und
- Ergebnis der Untersuchung ist bekannt/veröffentlicht.

Komplette Abdeckung der Möglichkeiten der Papier-Handzettel Wichtig für die sinnvolle Nutzung eines elektronischen Laufzettels ist, dass dieser mindestens die Möglichkeiten eines papierbasierten Laufzettels komplett abdeckt. Sind diese Mindestbedingungen nicht erfüllt, führt das dazu, dass die Anwender doch wieder eine Papierliste führen.

4.2 Anforderungen an die Hardware

Eine im Krankenhaus – speziell während der Visite – einzusetzende Software stellt bestimmte Anforderungen an die Hardware. Diese sind im Folgenden aufgeführt.

Größe und Gewicht Ein auf der Visite mitzuführendes Gerät sollte nicht größer als ein DIN-A4-Blatt sein. Größe und Gewicht sind beides Faktoren, die darüber

entscheiden, wie gut ein Tablet verwendet werden kann. Gleichzeitig sollte es so groß sein, dass zum einen Applikationen übersichtlich darstellbar sind. Zum anderen benötigen Darstellungen eine Mindestgröße, so dass diese von mehreren Personen gleichzeitig einsehbar sind.

Blickwinkel Der Bildschirm des Gerätes sollte so blickwinkelstabil sein, dass es von mehreren Personen zugleich betrachtet werden kann.

Hygiene Die Verwendung im medizinischen Umfeld erfordert es, dass das Gerät einfach zu desinfizieren ist.

4.3 Daten

Im Hinblick auf das spätere Design der Oberflächen ist es wichtig, bereits im Vorfeld zu wissen, welche Daten im System benötigt werden und dargestellt werden müssen. Die Wichtigkeit der verschiedenen Patienteninformationen wird auf jeder Station anders bewertet. Aus diesen Informationen wurden die in Abbildung 4.3 gezeigten Kategorien ermittelt, in die die Daten fallen.

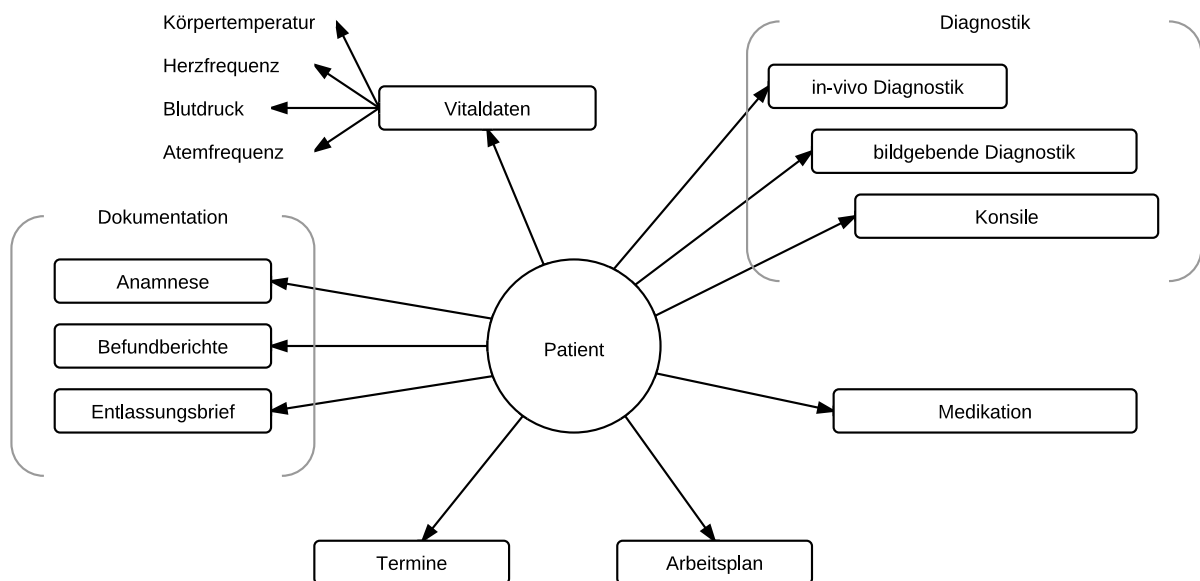


Abbildung 4.3: Modell der Patientendaten

Zu Vitalparametern werden Blutdruck, Herzfrequenz, Körpertemperatur und Atemfrequenz gezählt. Zusätzlich ist für manche Stationen der Blutzucker sowie die Sauerstoffsättigung interessant. Diese Daten werden in der Regel dreimal am Tag manuell vom Pflegepersonal genommen und in die Patientenkurve eingetragen.

Die Anamnese ist die Krankengeschichte aus Sicht des Patienten. Diese wird bei der Aufnahme des Patienten in das Krankenhaus erfragt. Die Gesamtanamnese setzt sich aus verschiedenen Teildaten zusammen. Dabei sind die wichtigsten Daten der

Grund des Besuchs der Klinik, eventuelle Vorerkrankungen, frühere sowie die aktuelle Medikation und bisherige Behandlungen.

Neben diesen Daten hat der behandelnde Arzt die Möglichkeit, verschiedene Untersuchungen anzuordnen, um zu einer Diagnose zu kommen. Diese Untersuchungen werden unter dem Begriff Diagnostik zusammengefasst und lassen sich in drei Kategorien einteilen: in-vivo-Diagnostik, bildgebende Diagnostik sowie Konsile. In-Vivo-Diagnostik umfasst Untersuchungen von lebendem Material wie Blut oder Urin, während bildgebende Diagnostiken beispielsweise Röntgen- oder CT-Untersuchungen sind. Als Konsil wird die Beratung eines Arztes durch einen Facharzt bezeichnet.

4.3.1 Zugriffe

Wie schon oben ausgeführt, werden zwei verschiedene Rollen Zugriff auf das System benötigen: Ärztliches Personal und Pflegepersonal. Diese beiden Rollen brauchen bei Ihrer Arbeit verschiedene Daten des Patienten. Neben den Daten ist auch die Art des Zugriffs zu beachten (siehe hierzu Tabelle 4.1). Da jeder Anwender möglichst nur die Informationen sehen sollte, die er für seine Arbeit benötigt, muss für das Pflegepersonal eine eigene Sicht geschaffen werden.

Tabelle 4.1: Zugriffe auf Patientendaten (l = lesend, s = schreibend)

Daten	Arzt		Pflege	
	l	s	l	s
Vitaldaten	✓		✓	✓
Medikation	✓	✓	✓	
Berichte	✓	✓		
Termine	✓	✓	✓	(✓)
Aufgaben	✓	✓	✓	(✓)
Diagnostik	✓			

4.4 Szenarien

Bei der Konzeption von Software bietet es sich an, Szenarien auszuarbeiten, die für die Benutzung typische Vorgänge und Umstände beschreiben. Diese müssen mit den Benutzern verifiziert werden. In Tabelle 4.2 auf der nächsten Seite werden die Szenarien beschrieben, die das *MEDo*-System unterstützen sollte.

4.5 Zusammenfassung

In diesem Kapitel wurden die Anforderungen an *MEDo* vorgestellt. Ausgangspunkt dafür waren die Arbeitsabläufe, die im vorhergehenden Kapitel beschrieben wurden. Dabei war es wichtig zu erkennen, dass die Arbeitsweisen der Ärzt/innen/e weiterhin

Tabelle 4.2: Szenarien

Szene	Beschreibung
1	Eine Ärztin/ein Arzt möchte eine Röntgenuntersuchung für einen Patienten machen. Er fängt mit dem Ausfüllen der Details an, wird dabei aber unterbrochen. Er füllt den Rest aus, sobald er Zeit findet.
2	Während einer Visite soll überprüft werden, ob bereits eine Röntgenuntersuchung angeordnet ist und wie deren Status ist.
3	Am Anfang einer Schicht möchte eine Ärztin alle neu eingetroffenen Untersuchungsergebnisse überprüfen, um auf den neuesten Stand zu kommen.
4	Vor der Visite bereitet sich eine Ärztin/ein Arzt auf die Visite vor. Dazu benötigt er die Anamnese, Labordaten, Röntgenaufnahmen sowie die Medikation.
5	Nach einer Visite möchte eine Ärztin/ein Arzt alle Aufgaben des heutigen Tages angezeigt bekommen. Er arbeitet diese nun nacheinander ab.
6	Während einer sollen viele Aufgaben nacheinander erstellt werden. Eine Ärztin/ein Arzt gibt dabei die Anordnungen, ein zweiter schreibt sie mit.

unterstützt werden muss. Ansonsten droht ein Rückfall auf Papierlaufzettel. Weiterhin wurden die Patientendaten, die im System unterstützt werden müssen, identifiziert. Um diese Anforderungen in der Praxis testen zu können, wurden mehrere Szenarien erstellt, die der Prototyp unterstützen sollte. Diese Anforderungen dienen als Ausgangspunkt für den im nächsten Kapitel vorgeestellten Entwurf des *MEDo*-Systems.

5

Entwurf

Ein Prototyp ist eine „Darstellung der Gesamtheit oder eines Teils eines interaktiven Systems, die, gegebenenfalls mit Einschränkungen, zur Analyse, Gestaltung und Bewertung verwendet werden kann“ [31]. Prototypen lassen sich in verschiedene Detailgrade einteilen. Am Anfang des Entwicklungsprozesses stehen meistens Papierprototypen. Diese fallen unter den Begriff *low-fidelity*. Auf der anderen Seite stehen *high-fidelity*-Prototypen. Diese ähneln auch funktionell schon stark dem Endprodukt – beispielsweise durch die Entwicklung im Zielframework. Beide Arten liefern bei der Evaluation ähnliche Ergebnisse [55, 57].

Es kann zwar von ersten Erfahrungen mit Smartphones ausgegangen werden, jedoch werden viele Anwender noch kein Tablet benutzt haben. Durch diese neue Form der Interaktion sind neue Interaktionskonzepte nötig. Viele erlernte Vorgehensweisen an Desktopcomputern sind bei Touch-Geräten nicht mehr möglich. Aus diesem Grund ist es wichtig, dass der Prototyp direkt auf einem Gerät getestet werden kann. Ich habe mich deshalb für einen *high-fidelity* Screen-Prototyp entschieden.

Wesentlich dabei ist vor allem, einen möglichst kompletten horizontalen Durchstich durch das System präsentieren zu können (siehe dazu Abbildung 5.1). Nicht beachtet wurden die verschiedenen Informationssysteme, in denen sich die Daten in der Realität befinden. Im Gegensatz zur fühlbaren Funktionalität stellen Hintergrundfunktionen keinen großen Faktor für einen Usability-Test dar.

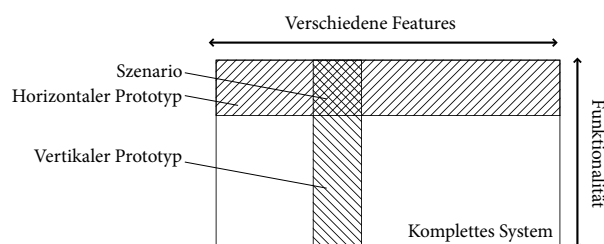


Abbildung 5.1: Unterscheidung zwischen horizontalen und vertikalen Prototypen [39, S. 94]

Während der Planung wurde bereits klar, dass im System zwei Sichten eine Rolle spielen: Auf der einen Seite gibt es die datenzentrierte Sicht, die Patientendaten, Befunde, Röntgenbilder und so weiter umfasst. Zum anderen hat das System eine

ablaufzentrierte Sicht, die Handlungen wie Aufgaben anlegen, Anforderungen verfolgen, neu eingetroffene Befunde anzeigen etc. einschließt. Obwohl bei dieser Arbeit letztere Sicht im Vordergrund steht, kann die datenorientierte Sicht nicht außer Acht gelassen werden. Erst durch die nahtlose Interaktion zwischen beiden Sichten kann der Benutzer optimal unterstützt werden. Insbesondere da der Prototyp möglichst realitätsnah die Benutzung eines mobilen Systems widerspiegeln sollte, musste im Hinblick auf die Evaluationsphase möglichst jeder Bildschirm zumindest dargestellt werden. So kann der Anwender sich besser vorstellen, wie das Gesamtsystem arbeitet und wird gedanklich nicht aus seiner Aufgabe gerissen. Es sollten nach Möglichkeit alle Interaktionen, die in den Szenarien (Abschnitt 4.4 auf Seite 35) definiert wurden, unterstützt werden.

Nachdem im vorangegangenen Kapitel alle Anforderungen erfasst wurden, müssen diese in Benutzeroberflächen umgesetzt werden. In diesem Fall wurde besonders darauf geachtet, dass das Tablet während der Visite möglichst in nur einer Hand gehalten werden muss. Dies ist am einfachsten, wenn das Gerät hochkant gehalten wird. Aufgrund der einfachen Umsetzung wird allerdings auch die Quer-Orientierung unterstützt. Dies ist sinnvoll, wenn beispielsweise Bilder oder Diagnoseergebnisse angezeigt werden sollen, die ansonsten nur knapp auf den Bildschirm passen.

5.1 Informationsarchitektur

Bevor irgendwelche Details zu den verschiedenen Bildschirmen entwickelt werden konnten, war es wichtig, einen Überblick über die Software zu bekommen. Es wurde ein Modell entwickelt (Abbildung 5.2 auf der nächsten Seite), das alle Bildschirme und deren Navigationsmöglichkeiten umfasst. Auf diese Weise konnte, bevor Details überlegt wurden, passende Navigationskonzepte für die verschiedenen Bereiche ausgewählt werden. Links oben ist der Ärzteschirm (auch Stationsschirm oder -übersicht) zu sehen. Hier werden Übersichten angezeigt, die alle Patienten betreffen – zum Beispiel eine Liste aller Aufgaben, Patienten und Untersuchungsergebnissen. Dies kann als Einstiegspunkt genutzt werden, um beispielsweise zum Schichtanfang eine Übersicht über die Station zu bekommen. Von hier kann der Benutzer außerdem einzelne Patientenakten auswählen.

Wählt er eine Aufgabe, sollte diese in einem Popover¹ oder einem modalen Fenster angezeigt werden. In dieser Aufgabenansicht sollten Details wie Ersteller, Datum, Status und natürlich der Text angezeigt werden. Zusätzlich sollten Aktionen angeboten werden, wie beispielsweise die Aufgabe als erledigt zu markieren. Andere Aktionen sind abhängig vom Kontext: So sollte bei einer Erinnerung, dass noch eine An-

¹Popover: Kleines, nicht modales Fenster, das immer einem Steuerelement (bspw. einem Button) durch einen Pfeil zugeordnet ist und durch Tippen ausserhalb dieses Elements ausgeblendet werden kann. Dieses Bedienelement wird durch das iOS-SDK auf dem iPad angeboten.

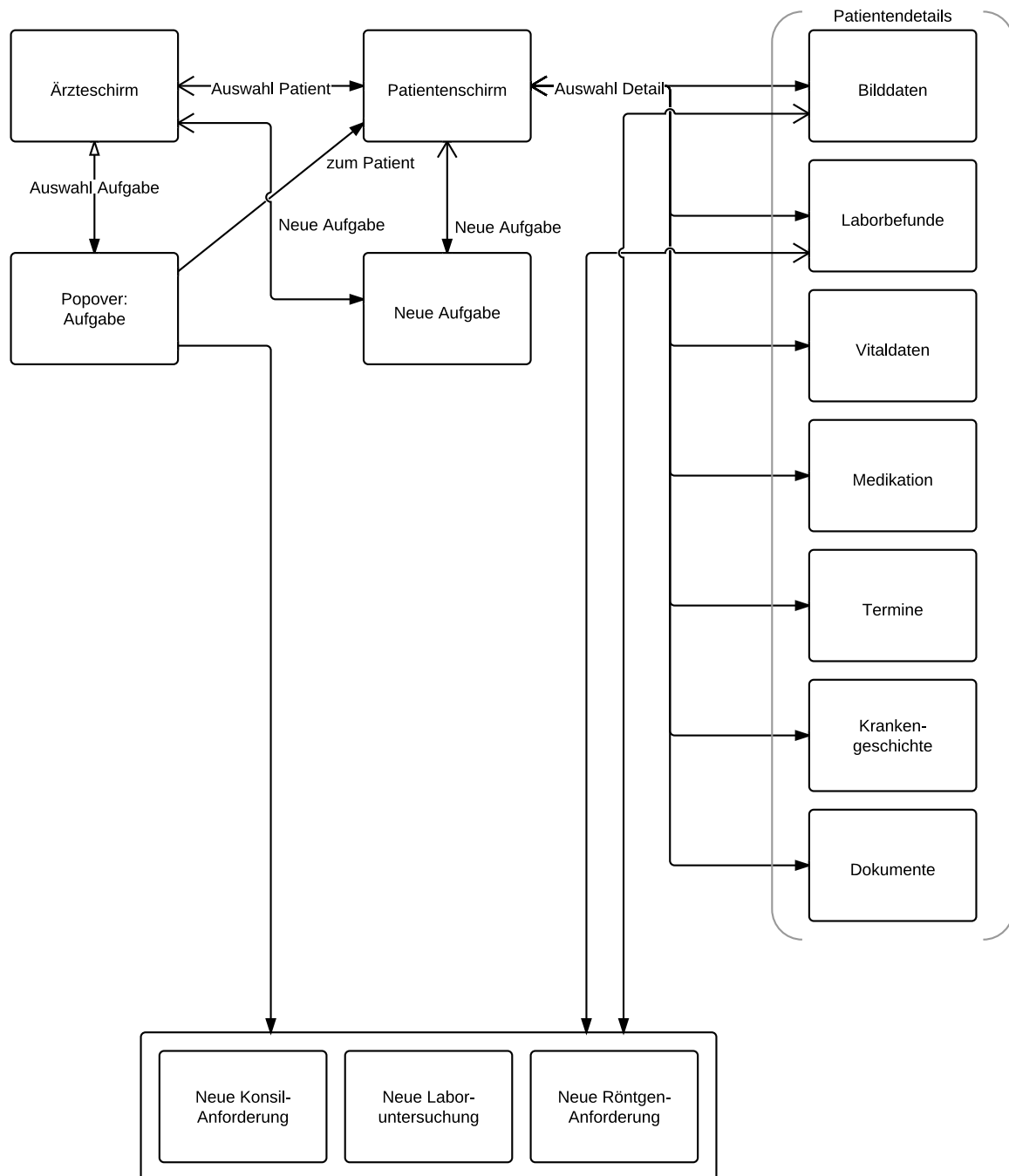


Abbildung 5.2: Informationsarchitektur für MEDo. Rechtecke stellen Bildschirme dar, Pfeile zwischen diesen Navigationsmöglichkeiten. Dabei geben die ausgefüllten Pfeile die Hauptrichtung an, einfache Pfeilspitzen zeigen an, dass hier eine Zurück-Funktionalität eingesetzt werden soll.

forderung vervollständigt werden muss, eine Möglichkeit (durch einen Button) geboten werden, um direkt zum entsprechenden Formular zu springen. Der Benutzer sollte auch die Möglichkeit haben, von der Aufgaben-Ansicht zum Patienten selber zu wechseln, um dort weitere Details zu erfahren.

Wieder auf dem Ärzteschirm kann der Benutzer ebenso zum Patient selbst springen. Hier sollte er neben den allgemeine Informationen Details angezeigt bekommen: Bilddaten, Laborbefunde, Vitaldaten, Medikation, Termine, Krankengeschichte und Dokumente. Für die meisten dieser Datenkategorien sollte eine eigene Ansicht eingeplant werden, da sonst der Bildschirm schnell überfüllt ist. Im Diagnostikbereich (Bilddaten und Laborbefunde) sollte es möglich sein, neue Anforderungen zu erstellen.

Generell sollte der Benutzer möglichst von jedem Bildschirm die Möglichkeit haben, eine neue Aufgabe zu erstellen. Insbesondere auf dem Stations- und auf dem Patientenschirm ist dies sehr wichtig, damit er nicht den aktuellen Kontext verlassen muss. Neue Anforderungen sollten je nachdem, um was für eine Untersuchung es sich handelt, angepasste Eingabemasken anzeigen. Beispielsweise benötigt eine Laboranforderung andere Daten als eine Konsilanforderung.

5.2 Navigationskonzept

Bei Systemen mit direkter Manipulation ist die Verteilung der Informationen und die Bewegung zwischen diesen von besonderer Bedeutung für die Interaktion. So sollte versucht werden, möglichst das räumliche Verständnis des Benutzers anzusprechen, indem Bildschirme auf einer zwei- oder dreidimensionalen Ebene angeordnet werden.

Angenommen es handelt sich um zwei aufeinanderfolgende Sichten, wie beispielsweise mehrere Seiten eines Suchergebnisses. Wechselt der Benutzer nun zum zweiten, rechten Bildschirm, so wird der linke Bildschirm nach links aus dem Bild geschoben, während der rechte von rechts erscheint (siehe Abbildung 5.3).

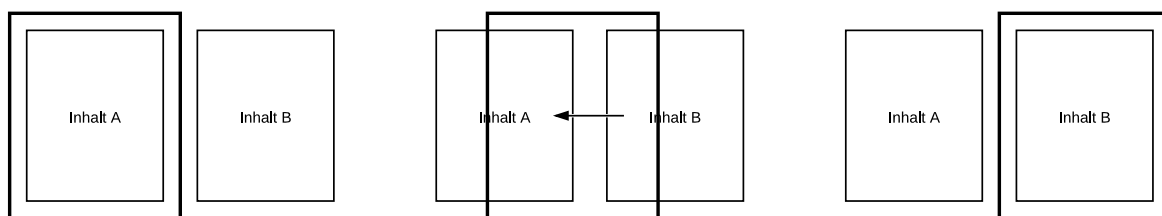


Abbildung 5.3: Projektion von Inhalten auf eine zweidimensionale Ebene. Inhalt A wird aus dem Bild animiert, während Inhalt B eingeschoben wird.

Durch solche Animationen kann das mentale Modell, welches sich der Benutzer von der Software macht, angesprochen werden. Diesem Usermodell gegenüber

steht das Implementierungsmodell, die der Entwicklungsseite inklusive der abstrakten Daten und Klassen entspricht. Der Entwickler versucht nun, über Repräsentierungsmodelle diese Details zu verstecken und gleichzeitig ein Modell anzubieten, welches eher dem Usermodell entspricht (siehe Abbildung 5.4). Ein Beispiel wurde oben bereits genannt, ein anderes wäre das animierte Einschieben einer neuen Tabellenzeile, wenn der Benutzer eine weitere Zeile einfügt.

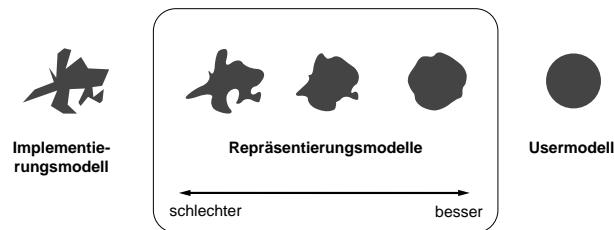


Abbildung 5.4: Es existieren qualitativ verschiedene Repräsentierungsmodelle zwischen Implementierungs- und Usermodell. Umso eher dieses dem mentalen Modell des Users entspricht, desto einfacher fällt ihm die Bedienung der Software [12, S. 59]

Im Fall von *MEDo* wurde besonders die Navigation zwischen Patienten und die Navigation zwischen den Patientendetails als Problempunkt erkannt, bei dem ein gutes Repräsentationsmodell wichtig ist. Abbildung 5.5 zeigt die drei in Erwägung gezogenen Modelle. Im ersten Modell sind die Patienten untereinander dargestellt und der Benutzer wechselt in horizontaler Richtung zwischen den Detailbildschirmen. Das zweite Modell ist um 90° gedreht, das heisst die Patienten sind vertikal untereinander angeordnet und die Patientendetails sind in horizontaler Richtung verteilt. Modell Nummer drei schließlich ist an letzteres angelehnt, wobei die Patienten hintereinander dargestellt werden, ähnlich der Organisation in einem Karteisystem. Dieses Konzept wurde schließlich im *MEDo*-System eingesetzt, abgewandelt durch den Einsatz von Reitern für die detaillierte Darstellung von Patientendetails. Die Gestaltung ähnelt damit der Patientenakte, bei der Seiten umgeblättert werden, wenn zwischen Patienten navigiert wird.

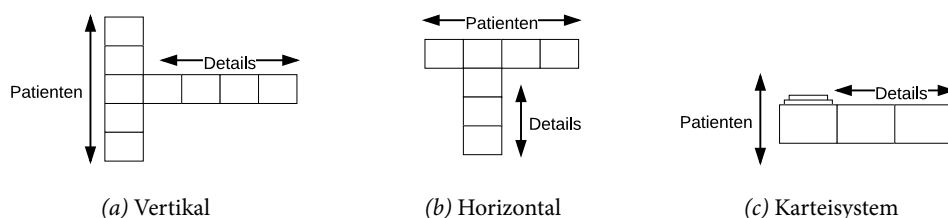


Abbildung 5.5: Repräsentationsmodelle für die Navigation zwischen Patienten und deren Daten

5.3 Screens

Die nächste Frage ist: Wie sollen die Informationen auf den verschiedenen Ansichten dargestellt werden? Hier müssen die Anforderungen an die Software und die Vorgaben der Plattform beachtet werden. Diese Entscheidungen werden in den folgenden Abschnitten vorgestellt.

5.3.1 Stationsschirm

Abbildung 5.6 zeigt Skizzen der Stationsübersicht. Hier findet sich der Anwender nach dem Start der Anwendung wieder. Abbildung 5.6a zeigt einen ersten Entwurf, bei dem die Patienten links dargestellt werden. Die rechte Seite kann genutzt werden, um dort Aufgaben oder, wie in Abbildung 5.6b, eine Karte der Station zu zeigen. Auf diese Weise kann der Benutzer sehen, welcher Patient in welchem Zimmer liegt. Bei der Visite kann diese Ansicht auch genutzt werden, um schnell von einem Raum zum nächsten zu wechseln. Dagegen spricht, dass mit der Auswahl des Zimmers ein Zwischenschritt eingeführt wird. Zusätzlich muss der Benutzer sich selbst den Grundriss der Station in Erinnerung rufen. Aus diesem Grund sollte alternativ eine Listenansicht verfügbar sein.

Abbildung 5.6c zeigt schließlich wie in dieser Stationsansicht eine einzelne Aufgabe ausgewählt werden kann. Die Darstellung erfolgt dabei in einem *two panel selector*. Die Aufgabenliste wird links, die Details einer ausgewählten Aufgabe rechts angezeigt. Bei der Übertragung dieser Skizzen auf ein iPad ist schnell ersichtlich, dass die Prototypen die Größe des Bildschirms nicht gut ausnutzen.

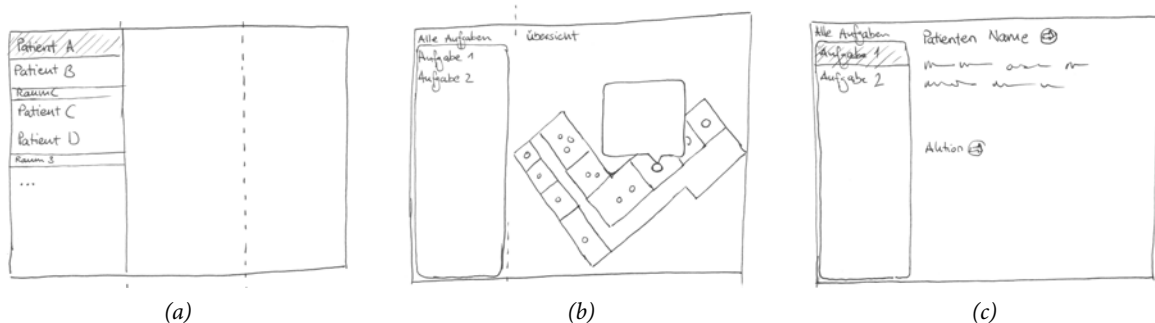


Abbildung 5.6: Verschiedene Aufteilungen der Stationsübersicht

5.3.2 Aufgaben erstellen und anzeigen

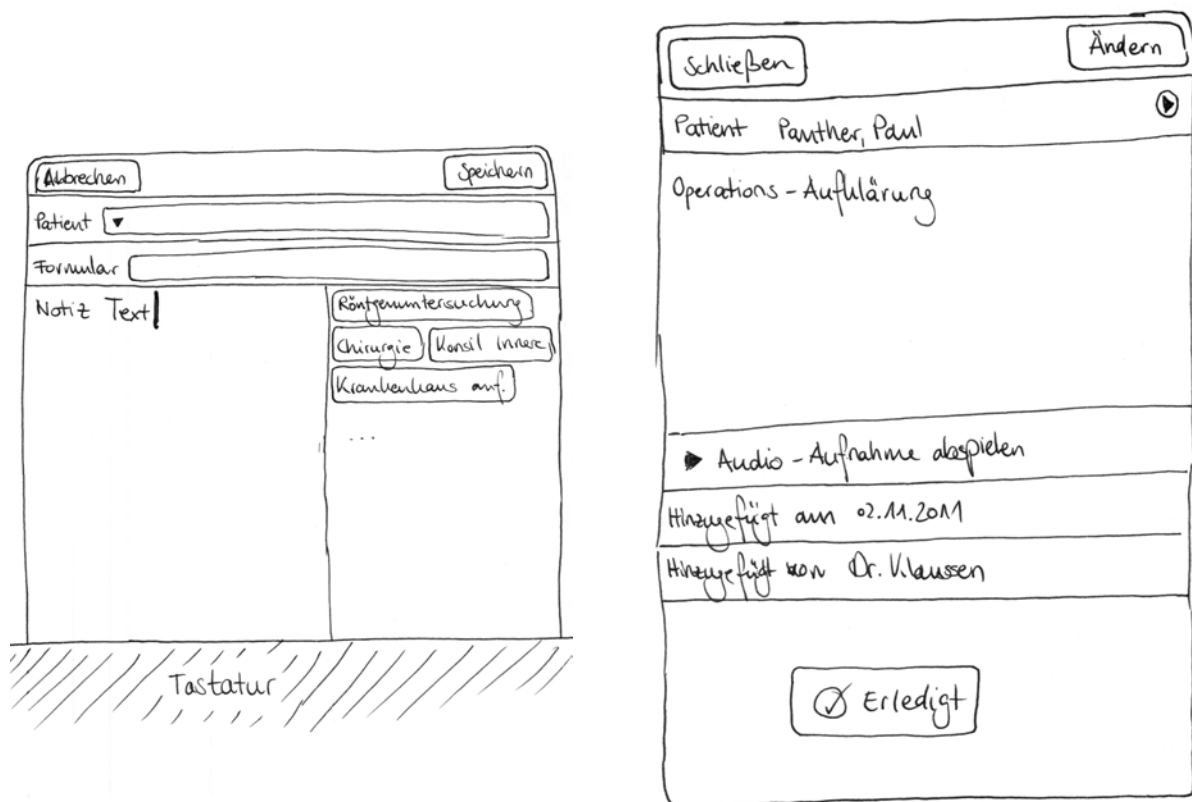
Das Erstellen von Aufgaben ist eines der wichtigsten Elemente im Prototyp. Die Eingabe sollte so schnell wie möglich geschehen. Deshalb habe ich mich für die Unterstützung durch Textbausteine entschieden. Diese können in einen Freitextbereich

eingefügt werden, der via Bildschirmtastatur editiert werden kann. So hat der Benutzer alle Freiheiten und kann gleichzeitig durch Textbausteine schneller Eingaben vornehmen.

Sie sollten je nach Station angepasst werden. Die Stationen sind zu unterschiedlich, als dass hier ein gemeinsamer Wortschatz angeboten werden kann. Es sollten allgemeine und oft gebrauchte Blöcke angeboten werden.

Abbildung 5.7a zeigt die vertikale Abfolge der Eingabefelder der Aufgabenanlege-Ansicht. Dadurch fällt es dem Auge leichter, dem Verlauf des Formulars zu folgen. Über der Tastatur schließlich befindet sich das Textfeld. In dieses kann entweder Freitext eingegeben werden oder komplette Textbausteine über den Bereich rechts davon eingefügt werden.

Zu beachten bei dieser Skizze ist, dass diese nur einen Ausschnitt aus dem gesamten Bildschirm zeigt. Die Ansicht ist als modales Fenster geplant, das nicht den kompletten Bildschirm bedeckt. Der nicht aktive Teil des Bildschirms sollte abgedunkelt erscheinen, damit er in den Hintergrund tritt. Wie bei allen iOS-Fenstern kann die Ansicht durch Buttons in einer Titelleiste gespeichert und abgebrochen werden.



(a) Fenster um eine neue Aufgabe anzulegen – Entwurf

(b) Fenster um Aufgaben anzuzeigen – Entwurf

Abbildung 5.7: Entwürfe zu Aufgabenfenstern

Die Anzeige von Aufgaben sollte in einem der Eingabe ähnlichen Fenster erfolgen. Abbildung 5.7b auf der vorherigen Seite zeigt einen Entwurf. Der Name des Patient wird an der gleichen Stelle angezeigt. Patientendetails können durch Antippen angezeigt werden, angedeutet durch einen Pfeil. Darunter finden sich alle Informationen zur Aufgabe sowie Buttons, um die Aufgabe als erledigt zu markieren und aufgenommene Audio-Aufnahmen abzuspielen.

5.3.3 Navigationsleiste

Ein interaktives System erhält durch ein visuelles Framework eine einheitliche Gestalt und trägt somit zur Wiedererkennung bei. Es umfasst Farben, Abstände, Ausrichtungen und Typografie. Zusätzlich sollten Navigationsmöglichkeiten und „Sie befinden sich hier“-Wegweiser auf allen Ansichten ähnlich sein [54].

Es sollte immer ersichtlich sein, wo sich der Benutzer gerade befindet und welche Möglichkeiten er hat, von hier aus zu navigieren. Beispielsweise sollten diese auf einem Patientenschirm sein:

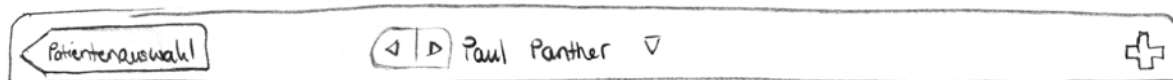
- Zurück zur Stationsübersicht,
- zwischen Patienten hin und her wechseln,
- direkt zu einem Patienten springen,
- sowie eine neue Aufgabe anzulegen.

In Abbildung 5.8 auf der nächsten Seite werden diese Anforderungen in verschiedener Art exploriert. Allen Skizzen gemein ist, dass der Patientennamen immer in der Navigationsleiste angezeigt wird. Das ist das normale Verhalten in allen iOS-Anwendungen. Genauso verhält es sich mit dem Zurück-Button oben links. Durch die Pfeilform ist die Richtung erkennbar, die der Benutzer mit der Einblendanimation dieses Bildschirms in Verbindung bringen kann. Der Pfeil zeigt entsprechend in die entgegengesetzte Richtung.

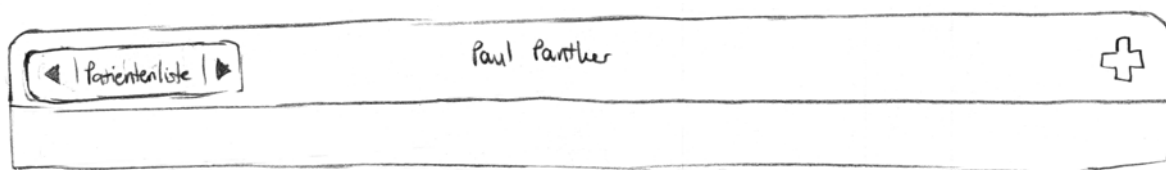
Die Navigationsleiste besitzt drei logische Unterteilungen. Links, in der Mitte zentriert, sowie rechts. Der linke Bereich ist bereits durch den Zurück-Button belegt, genauso wie mittig der Patientennamen angezeigt wird. Abbildung 5.8a fügt dem Patientennamen durch einen Dropdownpfeil noch weitere Funktionalität hinzu. Über diesen kann der Benutzer die Patientenliste aufrufen. Außerdem kann der Benutzer durch Buttons links vom Namen vor und zurück durch die Patienten schalten kann. Hier ist darauf zu achten, dass die Pfeilrichtungen den Übergangsanimationen zwischen den Patienten entsprechen muss, damit der Benutzer ein durchgängiges Mapping für sein mentales Modell aufbauen kann.

5.3.4 Patientenschirm

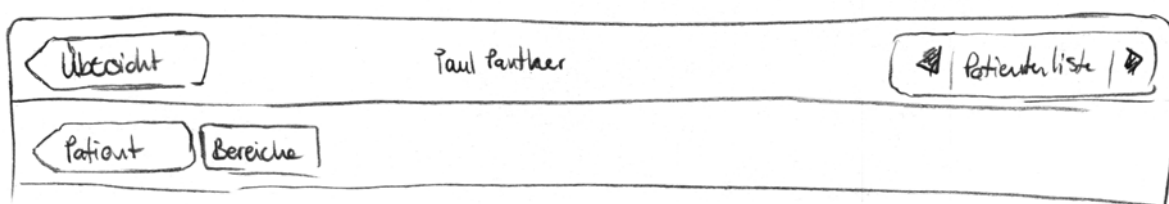
Der Patientenschirm ist der Ausgangspunkt zu allen Informationen, die einen bestimmten Patienten betreffen. Dazu wird zum einen eine Übersicht, zum anderen eine Möglichkeit, Detailinformationen abzurufen, benötigt.



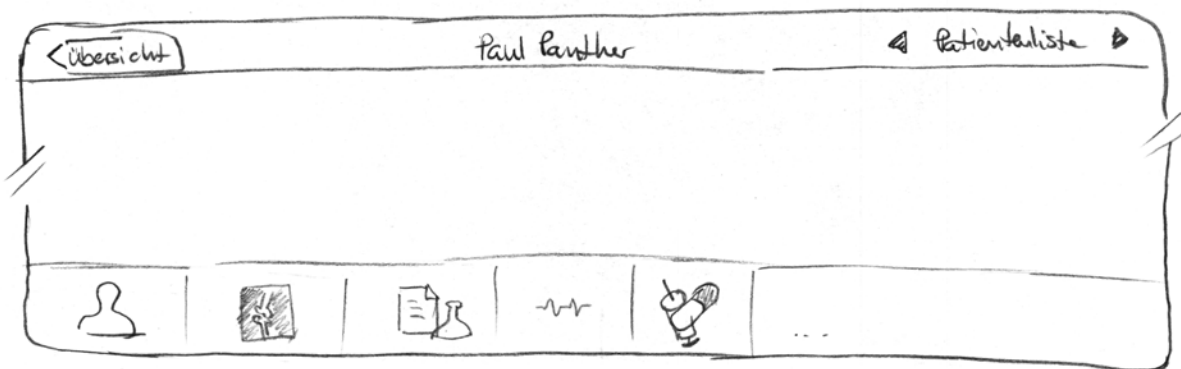
(a)



(b)



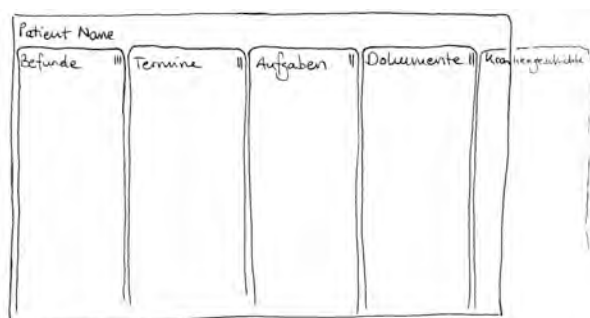
(c)



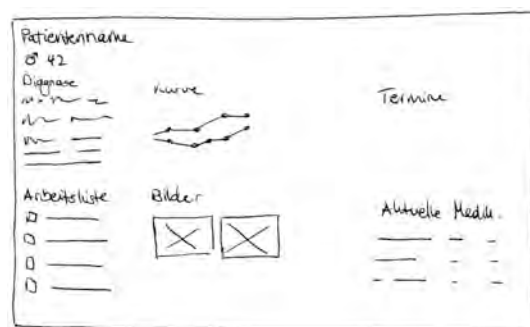
(d)

Abbildung 5.8: Navigationsleisten

Eine Idee wäre die Darstellung aller Informationen in Spalten, wie in Abbildung 5.9a dargestellt. Das heißt es gibt eine Spalte für Termine, eine Spalte für Aufgaben und weitere Spalten entsprechend den Datenkategorien. Da nicht alle diese Spalten auf einen Bildschirm passen, kann diese Ansicht durch horizontales Wischen bewegt werden. Um mehr Einträge zu einer Kategorie einsehen zu können, scrollt der Benutzer nach unten.



(a) Darstellung der Patientendaten in scrollbaren Spalten



(b) Alternative Anordnung in einem Raster

Abbildung 5.9: Entwürfe zur Darstellung der Patientendaten

Denkbar wäre, die Übersicht nach einem Raster auszurichten (siehe dazu Abbildung 5.9b). So braucht der Benutzer nicht nach den Informationen zu suchen, sondern er bekommt die Wichtigsten auf einem Bildschirm angezeigt.

Da bei dieser Anordnung nicht alle Einträge zu allen Kategorien angezeigt werden können, werden weitere Bildschirme benötigt. Um zwischen diesen umschalten zu können, gibt es im iOS-Framework eine Tab-Leiste. Diese sind in iOS standardmäßig am unteren Bildrand angeordnet (wie in Abbildung 5.10 auf der nächsten Seite). So kann der Benutzer auf dem ersten Reiter einen Überblick bekommen, und für mehr Informationen weitere Seite anwählen.

Abbildung 5.10 zeigt aber auch, dass die Tab-Leiste keine visuelle Verbindung zum Navigationsbereich oben hat. Es ist schwierig zu sagen, welchen Teil der Ansicht die Tab-Leiste kontrolliert: Ändert sich die Navigationsleiste, wenn auf einen anderen Reiter umgeschaltet wird? Oder bleibt sie gleich und nur der Inhalt ändert sich?

Gelöst habe ich dieses Problem durch die Positionierung der Tableiste am oberen Bildschirmrand, unter der Navigationsleiste (siehe Abbildung 5.11 auf Seite 48). Durch die logische Gruppierung des Namens und den Tabreitern wird eine visuelle Einheit gebildet, die ähnlich einer Akte mit Karteireitern ist.

5.4 Styleguide: Piktogramme, Farben und visueller Stil

Die Erlernbarkeit eines interaktiven Systems kann durch einen einheitlichen visuellen Stil unterstützt werden. Nicht passende, visuelle Elemente werden vom Benutzer als störend empfunden.

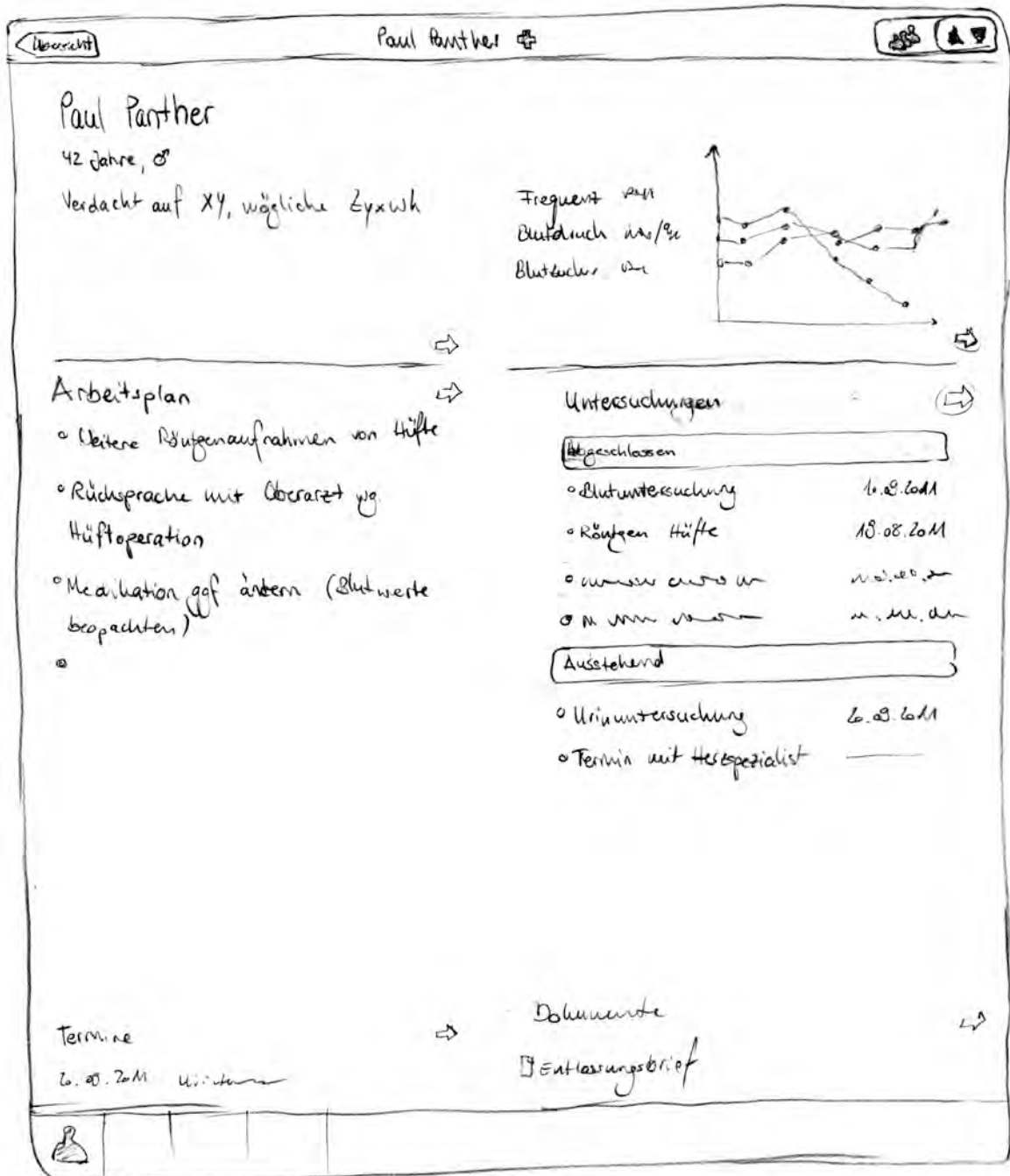


Abbildung 5.10: Patientendaten in Portratorientierung im Raster, mit Tab-Leiste

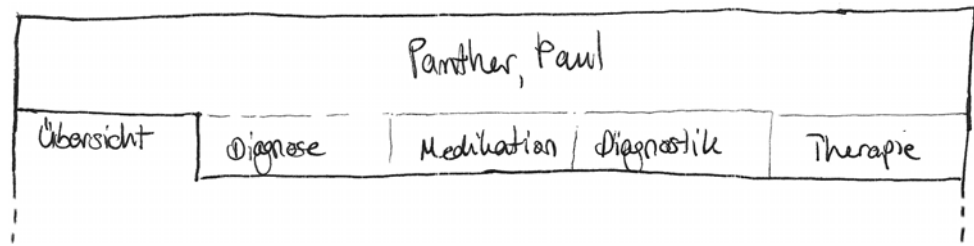


Abbildung 5.11: Tab-Leiste am oberen Rand

Ein minimaler Stil war das Ziel für *MEDo*. Dabei entsteht hierarchische Ordnung durch Typografie und Struktur durch Leerräume. Farben werden nur sparsam eingesetzt. Nur aktive Elemente wie Selektionen oder gedrückte Buttons werden farblich gekennzeichnet und werden so besonders hervorgehoben. Abbildung 5.12 zeigt einen Entwurf eines Buttons. Die Dreidimensionalität lässt ihn als Button erkennen, besonders durch die zurückhaltende Gestaltung des restlichen Interfaces. Er soll eine hellgraue Oberfläche haben und aufleuchten, wenn er gedrückt wird.



Abbildung 5.12: Entwurf eines Buttons

Abbildung 5.13 zeigt verschiedene Pixel-Designs der Buttons. Diese werden besonders in Navigationsleisten sowie Toolbars eingesetzt. Der gleiche Stil wird außerdem bei segmentierten Steuerelemente genutzt.

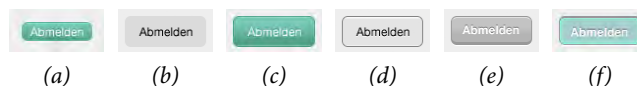


Abbildung 5.13: Verschiedene Pixeldesigns der Buttons

Steuerelemente benötigen eine bestimmte Mindestgröße, damit der Benutzer sie zielsicher treffen kann – egal ob Maus- oder Touchbedienung. Da Finger aber signifikant größer sind als Mauszeiger, müssen die Zielbereiche bei direkter Manipulation besonders groß sein. Da intern Pixel verwendet werden, muss eine Umrechnung vom physikalischen Wert stattfinden. Maßgebend dafür sind die DPI (Dots per Inch) des Gerätes. Dieser Wert gibt das Verhältnis von physischen Einheiten (Inch) zur Geräteauflösung in Pixeln an. Typische Druckerzeugnisse haben 300 Dots per Inch (DPI), Computer-Bildschirme um die 100 DPI. Teilweise wird auch von PPI (Pixel per Inch) gesprochen, beides meint aber im Endeffekt das gleiche.

Apple schlägt bei den 163 DPI des iPads eine Mindestgröße von 44×44 Pixeln vor [6].

5.4.1 Typographie & Raster

Als Grundtype soll im System „Helvetica Neue“ eingesetzt werden. Text soll im Schnitt Normal in 15 Punkt gesetzt werden. Für Hauptüberschriften (Navigationsleiste, Name des Patienten, etc) ist Fein/20pt zu verwenden, für Überschriften der zweiten Ordnung Schmall/Fett/15pt. Manchmal wird etwas kleinerer Text für zusätzliche Informationen o.Ä. benötigt, dieser wird dunklgrau in 12pt gesetzt.

Mithilfe eines Raster – bestehend aus Seitenrändern, Spalten und Zwischenräumen – lassen sich einfach Strukturen bilden und so eine visuelle Ordnung herstellen. Innerhalb des *MEDo*-Systems sollte bei der Positionierung von Elementen ein Raster, bestehend aus zwölf Spalten, verwendet werden (siehe Abbildung 5.14). Der Vorteil bei einem Raster bestehend aus zwölf Spalten ist, dass sich einfach darauf aufbauende Raster bilden lassen. So sind Raster mit sechs, vier, drei und zwei Spalten möglich. Bei Portraitorientierung des Gerätes sind horizontal 768 Pixel verfügbar. Hier bietet sich ein sechser-Raster an, wobei sich 114 Pixel breite Spalten bei 10 Pixel Seitenrändern und Spaltenzwischenräumen ergeben.



Abbildung 5.14: Raster

5.4.2 Farben

Es sollte nur eine primäre Farbe eingesetzt werden: Ein türkis-grün, die Farbe von Operationskleidung im Krankenhaus. Einige verschiedene Versuche waren notwendig, um die richtige Farbe und Sättigung zu finden. Denn neben dem Farbton sollte noch eine Variation durch eine andere Sättigung gefunden werden. Letztendlich werden in *MEDo* die Farben (19,201,142) und (15, 155, 127) (RGB) eingesetzt (siehe Abbildungen 5.15c und 5.15d).



Abbildung 5.15: Farben

5.4.3 Piktogramme

Das iOS-Software Development Kit (SDK) bietet einige integrierte Icons an. Diese sind minimalistisch, zweidimensional und einfarbig. Dem Entwickler steht natürlich frei andere Icons einzusetzen, aber die meisten Komponenten, die Icons anzeigen bzw. unterstützen (`UITabBar`, `UIBarButtonItem`), konvertieren automatisch den Alpha-wert in den Weiß-Kanal. Das heißt dass Icons, die diesen Komponenten zugewiesen werden, nur als Umrisse dargestellt werden. Alle Farbinformationen gehen verloren.

Die Piktogramme im *MEDo*-System sollten ähnlich einfach sein. Da die Evaluierung an ungeübten Usern geschehen sollte, erschien es am sinnvollsten, möglichst keine Piktogramme ohne Beschriftung einzusetzen. Wiedenbeck [62] zeigte, dass Erstbenutzer erheblich länger brauchen, um die Bedeutung der Symbole zu verstehen. Diese Lerneffekte stellen sich nach Lansdale [33] vor allem durch die Position ein, weniger durch das verwendete Symbol selber. Im System sollten somit Symbole nur eingesetzt werden, wenn zusätzlich eine Beschriftung angebracht ist oder wenn die Funktionalität auch auf anderem Wege zu erreichen ist.

Abbildung 5.16 auf der nächsten Seite zeigt mehrere Iconentwürfe. Abbildung 5.16b zeigt drei Möglichkeiten, wie die Patientenliste symbolhaft dargestellt werden kann. Die beiden letzten Icons zeigen durch drei Balken den Listencharakter an. Diese Balken stehen zwar in vielen Anwendungen für eine Liste, diese könnte sich aber auch auf einen einzelnen Patienten beziehen.

Abbildung 5.16d stellt verschiedene Icons dar, mit denen Diagnostika bzw. Medikation dargestellt werden können. Röntgenbilder werden durch eine weiße Knieknochenkontur dargestellt. Charakteristisch ist außerdem, dass die weißen Knochen von hellem Gewebe umgeben sind. Als Referenz haben hier verschiedene Röntgenbilder gedient. Eine Kombination von einer Spritze und einer Medikamentenkapsel wurde als Piktogramm für Medikation entwickelt. Ein *Erlenmeyerkolben* oder ein *Reagenzglas* in Kombination mit einem Dokument soll einen Laborbericht darstellen.

5.5 Zusammenfassung

Im Verlauf dieses Kapitels wurden alle Screens, die in *MEDo* dargestellt werden, spezifiziert. Dazu wurde eine einheitliche visuelle Sprache gefunden, die Abstände, Typografie, Raster, Farben, Piktogramme und Steuerelemente umfasst. Aufgrund dieser Spezifizierungen konnte nun die Implementierung beginnen. Diese wird im nächsten Kapitel beschrieben.

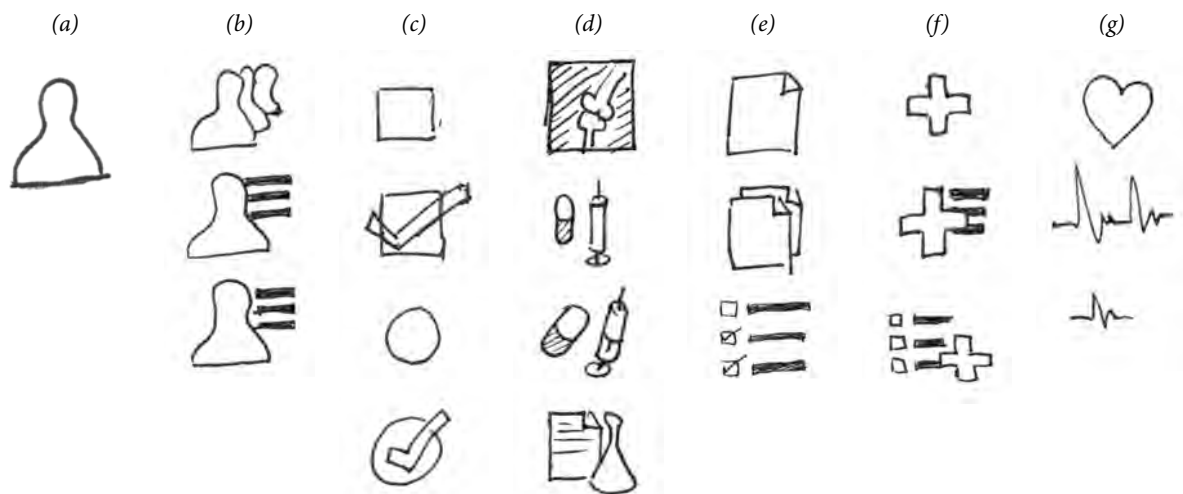


Abbildung 5.16: Entwürfe für verschiedene Icons

6

Implementierung

Objective-C ist die einzige Programmiersprache, mit der native Anwendungen für iOS, dem Betriebssystem von Apple iPhone, iPad und (angepasst) auch Apple TV geschrieben werden können [19, 47]. Es wäre theoretisch auch die Entwicklung einer Webanwendung denkbar gewesen. Da dabei oft die Geschwindigkeit leidet und das System nicht offline laufen könnte, wurde diese Idee verworfen.

Objective-C, wie dem Namen bereits zu entnehmen ist, setzt auf die Programmiersprache C auf und erweitert diese um objektorientierte Konzepte. Die Geschichte von Objective-C reicht zurück bis zum NeXTSTEP-Betriebssystem, das 1989 veröffentlicht wurde. Heutzutage wird es neben der Entwicklung für iOS-Geräte auch für Mac OS X genutzt.

Bei der iOS-Programmierung wird auf das Cocoa Touch API (Application Programming Interface) zurückgegriffen. Durch Cocoa Touch werden Hardware- und Softwarefunktionen abstrahiert. Beispiele von Funktionen, die durch Cocoa Touch angeboten werden, sind hardwareunterstützte Animationen, Multitasking und Gestenerkennung [21, 49].

Apple bietet ein SDK an, mit Hilfe dessen Anwendungen geschrieben werden können. Darin enthalten ist mit Xcode eine Integrated Development Environment (IDE), die allerdings nur für OS X verfügbar ist [19, 47].

Im SDK ist ein Simulator enthalten, der die Ausführung des Codes auch auf i86 Geräten ermöglicht. Auf echten Geräten kann Software nur mit einer kostenpflichtigen Mitgliedschaft im iOS Developer Programm getestet werden. Durch diese Mitgliedschaft kann man außerdem die entwickelten Applikationen im *App Store* anbieten, durch den Kunden auf allen iOS-Geräten Applikationen kaufen und installieren können [47].

Cocoa Touch enthält eine Anzahl von Frameworks, die bestimmte Funktionalitäten anbieten, wie z.B. Karten-Funktionen, Werbeeinblendungen und Spieleunterstützung. Das wichtigste Framework neben Foundation Kit, welches Basisklassen enthält, ist UIKit. Es bietet Klassen, um Benutzeroberflächen in Anwendungen zu erstellen. Dazu gehören Ereignishandling, Steuerelemente und Verwaltung der Darstellungshierarchie.

Cocoa Touch folgt dem Model View Controller (MVC) Muster. Es handelt sich da-

bei um ein Softwarearchitekturmuster, bei dem Applikationslogik von Benutzereingabe und -ausgabe getrennt wird. Das *Model* enthält die Daten der Software, teilweise auch die Geschäftslogik. *Views* sind alle Komponenten, die die Daten darstellen und Eingaben entgegennehmen. *Controller* übernehmen schließlich die Verarbeitung der Eingaben und reagieren auf diese, beispielsweise durch Ändern der *View*.

Je nach Programmiersprache wird das MVC-Muster unterschiedlich angewandt. Dabei ist das Hauptunterscheidungsmerkmal, ob die *View* direkt oder nicht direkt auf die Daten im *Model* zugreift. Abbildung 6.1b zeigt, dass die *View* bei Cocoa keinen Zugriff hat (im Gegensatz zu Java, Abbildung 6.1a), sondern vom *Controller* mit Daten versorgt wird.

Im Folgenden wird die Implementierung eingeteilt in die Komponenten *Model*, *View* und *Controller* dargestellt.

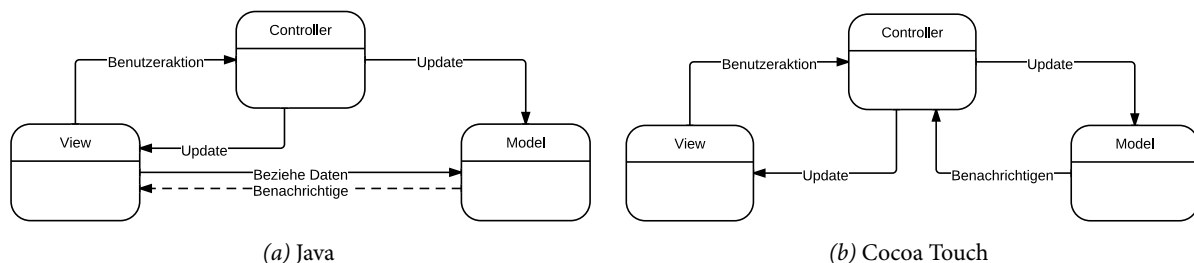


Abbildung 6.1: Model View Controller

6.1 Model

Das *Model* besteht in vielen Fällen hauptsächlich aus einer Anbindung an eine Datenbank. So verhält es sich auch bei *MEDo*. Die Anbindung an externe Systeme wie beispielsweise die jeweiligen KISSs wäre sehr zeitintensiv gewesen. Da möglichst viele Stationen in unterschiedlichen Krankenhäusern angesprochen werden sollten, wurde auf die Entwicklung entsprechender Schnittstellen verzichtet. Stattdessen wurde eine Datenbank aufgesetzt, welche mit Demodaten gefüllt ist. So erhält der Benutzer den Eindruck, dass es sich um ein aktives System handelt.

Im Hintergrund läuft sich eine SQL-Datenbank (sqlite). Diese wird durch das Core Data Framework angesprochen. Dieses Framework unterstützt den Programmierer dadurch, dass Persistenz und Funktionalitäten wie *Rückgängig* automatisch zur Verfügung stehen. Es bildet eine Schicht, so dass Modellbeschreibungen automatisch in Klassenbeschreibungen umgewandelt werden. Das Framework hält nur die nötigsten Daten im Speicher (*lazy loading*).

6.1.1 ER-Modell

Das Datenbankmodell lässt sich logisch in die drei Säulen Patientendaten, Aufgabenmanagement und Organisation/Arbeitsmittel einteilen. In den folgenden Abschnitten werden diese beschrieben. Abbildung 6.2 zeigt das UML-Datenbankmodell.

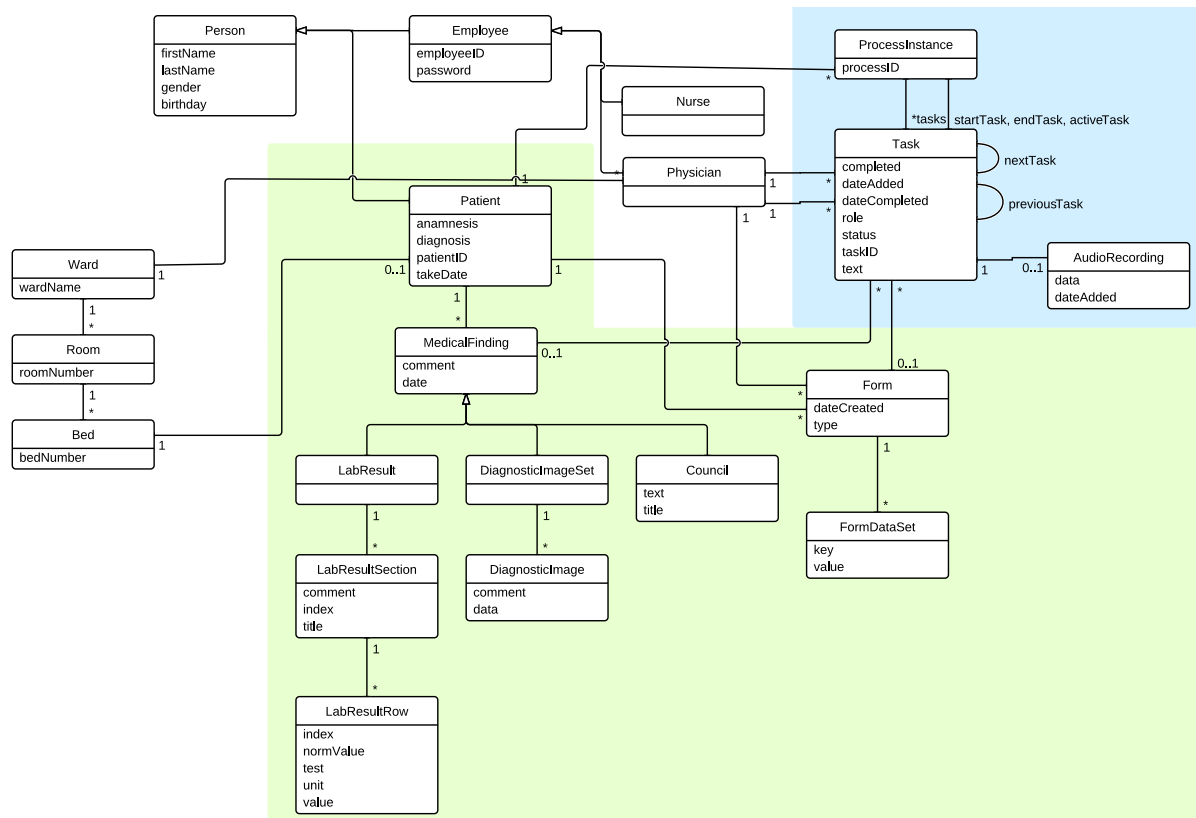


Abbildung 6.2: Datenbankaufbau in UML-Notation. Der Bereich Patientendaten ist grün hinterlegt, der Bereich Aufgabenmanagement blau. Organisation/Arbeitsmittel sind nicht hinterlegt.

Patientendaten

Zu den Patientendaten zählen die Basisdaten wie Name, Alter, Aufnahmedatum und Patientenidentifikation. Für die medizinische Informationsverarbeitung sind weiterhin die Anamnese, die sich weiter in Medikativ-, Sozial- und Familienanamnese unterteilen lässt, und die aktuelle Diagnose von Interesse. Darüber hinaus können Patienten Untersuchungsergebnisse zugeordnet haben. Davon sind Labor, bildgebende Diagnostik und Konsile im System implementiert.

Das Ergebnis einer Laboruntersuchung wird in `LabResult` gespeichert. Jedes `LabResult` besteht aus mehreren Teiluntersuchungen, die als `LabResultSections` modelliert sind. Diese haben einen Titel und enthalten mehrere `LabResultRows`, die den Test, das Ergebnis, die Einheit und erwarteten Normwert angeben.

Ein `DiagnostImagSet`, also beispielsweise ein Ergebnis einer Röntgenuntersuchung, enthält mehrere `DiagnosticImages`, die zum einen jeweils einen Kommentar haben können, zum anderen die binären Bilddaten enthalten. Im Demo-System sind dies PNG- sowie JPG-Dateien.

Ein Konsil wird als unformatierter Text im System gespeichert, der optional auch einen Titel haben kann.

Jedem Patient können auch Formulare (`Form`) zugeordnet werden. Diese bestehen aus Feldern (`FormDataSet`), die jeweils als Schlüssel-Wert-Paar gespeichert werden. Diese Datenstruktur macht es möglich, sehr verschiedene Formulare im System zu speichern. Je nach Formular-Typ kann das Formular dem Benutzer anders dargestellt werden. Der jeweilige Formularcontroller liest die Daten zu den zum Formular gehörenden Schlüsseln aus und stellt sie auf dem Formular dar, bzw. speichert sie ab.

Aufgabenmanagement

Nachdem mit den Patientendaten die datenzentrierte Sicht abgedeckt ist, müssen die Strukturen für die prozessorientierten Daten definiert werden. Die wichtigsten Elemente sind hier `ProcessInstance` und `Task`.

Aufgaben werden durch `Tasks` modelliert. Sie haben als Attribute einen Schalter, ob sie bereits abgeschlossen sind, einen Text, die Rolle, die die Aufgabe ausführen soll und das Datum an dem sie erstellt und abgeschlossen wurden. Weiterhin haben sie einen Mitarbeiter, der die Aufgabe abgeschlossen hat. Sie können außerdem eine Verknüpfung zu einem Formular oder einer Diagnostik haben. Aufgaben können eine Audioaufnahme assoziiert haben. Diese wird in einer zusätzlichen Tabelle (`AudioRecording`) gespeichert, damit die Daten nur geladen werden müssen, wenn explizit auf sie zugegriffen wird, um die Speicherverwendung gering zu halten.

Eine Prozessinstanz ist einem Patienten zugeordnet und hat einen Arzt oder Ärztin, der sie erstellt hat. Sie kann auf einer Prozessbeschreibung basieren (siehe dazu Abschnitt 6.4 auf Seite 76). Dies wird durch das Attribut `processID` gekennzeichnet. Ist es nicht gesetzt, so handelt sich um einen einzelnen Arbeitsschritt. Die Attribute `startTask`, `endTask` und `activeTask` geben an, welcher der Aufgaben, die in `tasks` aufgeführt sind, die erste, letzte und aktive Aufgabe ist. Dabei wird logischerweise `endTask` erst gesetzt, wenn bei dem Prozess die letzte Aufgabe abgeschlossen ist. Das Attribut `tasks` enthält nicht von Anfang an alle Aufgaben, sondern nur die aktuelle und bereits abgeschlossene. Es wäre also möglich, Flexibilität in den Prozessen zu unterstützen.

Organisation und Arbeitsmittel

Im System werden Stationen (`Ward`) mit Räumen und Betten modelliert. Ein Bett kann einem Patienten zugeordnet sein. Stationen, Räume und Betten haben jeweils eine Bezeichnung oder Nummer.

Bereits erwähnt wurden Mitarbeiter (Employee). Diese haben eine IS-A-Beziehung zu Person und fügen dessen Attributen Vorname, Nachname, Geschlecht und Geburtstag die Personalnummer und ein Kennwort hinzu. Auch Patient hat eine IS-A-Beziehung zu Person. Employee hat die Subklassen Physician (Arzt/Ärztin) und Nurse (KrankenpflegerIn), die zur Unterscheidung der Rollen im System dienen.

6.2 View

Das iOS SDK bietet seit Version 5.0 eine Möglichkeit an, alle Instanzen einer Steuerelementklasse zu ändern. Dazu steht dem Entwickler das `UIAppearance`-Protokoll zur Verfügung. Dieses Protokoll verstehen die meisten UIKit-Klassen. Die Verwendung ist sehr einfach: Anstatt beispielsweise den `setTintColor:` Befehl an eine Instanz zu schicken, muss dieser an den Proxy `appearance` der Klasse geschickt werden: `[[UIBarButtonItem appearance] setTintColor:[UIColor blackColor]]`.

Um dem Styleguide zu genügen, wurden fast alle Komponenten umgestaltet. Einige Beispiele sind in Abbildung 6.3 dargestellt. Für die beiden Komponenten `UISegmentedControl`, `UIBarButtonItem` und `UINavigationController` wird jeweils erst das Aussehen vor und nach der Anwendung der Änderungen gezeigt.

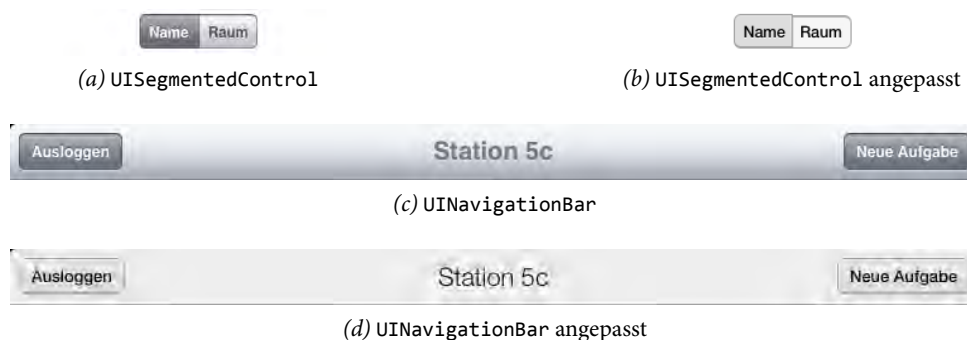


Abbildung 6.3: Angepasste Interface-Elemente

Durch diese leichte Anpassbarkeit war es nur sehr selten nötig, eigene Steuerelemente zu programmieren. Hervorzuheben ist das Steuerelement, welches es ermöglicht, Textbausteine in den Text einzufügen. Die Funktionsweise wird im folgenden Abschnitt erklärt.

6.2.1 `MDTextModuleListControl`

`MDTextModuleListControl` erbt von `UIControl` und stellt eine Liste von Strings durch Buttons dar (siehe Abbildung 6.4 auf der nächsten Seite). Diese Buttons fügen bei Berührung ihre Beschriftung in ein vorher angegebenes `UITextView` ein, wobei

diese Bewegung mit einer Animation begleitet wird. `UIControl` ist die Basisklasse für alle Steuerelemente, die nicht nur der Darstellung dienen (`UIView`), sondern auch auf Interaktionen des Benutzers reagieren. Es bietet damit eine einheitliche Schnittstelle für Steuerlemente wie Buttons und Slider. Diese Objekte unterstützen den *target-action*-Mechanismus, so dass Methoden für bestimmte Aktionen (wie Berühren und Wischen) registriert werden können.



Abbildung 6.4: `MDTextModuleListControl` nach Antippen des Buttons „Medikation“

Das öffentliche Interface der Klasse ist sehr simpel (siehe folgendes Listing). Es bietet ein Array für die Textbausteine und ein `UITextView`, in das die Werte eingefügt werden sollen.

```
@interface MDTextModuleListControl : UIControl
@property (nonatomic, strong) NSArray *values;
@property (nonatomic, strong) UITextView *textView;
@end
```

Zusätzlich besitzt die Klasse noch ein verstecktes Interface (siehe folgendes Listing), das Methoden enthält, die nicht öffentlich sind:

```
@interface MDTextModuleListControl (hidden)
- (void)updateButtons;
- (CGPoint)carretPositionInTextView:(UITextView *)textView;
- (void)didSelectTextModule:(NSString *)string
  inRect:(CGRect)sourceRect;
@end
```

Die Methode `(void)updateButtons` wird aufgerufen, wenn sich die `values`-Property ändert. Sie instanziiert und platziert die Buttons auf dem Steuerelement. Wenn ein Button nicht mehr in die aktuelle Zeile passt, rutscht dieser in die nächste Zeile.

Bei Berührung eines Buttons wird die Beschriftung des Buttons in das `UITextView` bewegt. Dafür wird ein temporäres `UILabel` mit gleicher Beschriftung erstellt und über dem Button platziert. Von dieser Position aus wird das Label nun zur Zielkoordinate, der Selektionsmarke im `UITextView`, bewegt. Nach Abschluß der Animation wird das Label entfernt und der Textblock im `UITextView` hinter der Selektionsmarke eingefügt.

Die Positionsbestimmung des aktuellen Textcursors innerhalb eines UITextView wird nur in der Form „Zeichen von X bis Y sind selektiert“ unterstützt. Leider ist keine direkte Abfrage der Pixelkoordinaten möglich. Aus diesem Grund wird die Methode (CGPoint) `carretPositionInTextView:` benötigt. Sie versucht zu schätzen, welche Koordinaten der Textcursor im angegebenen UITextView hat.

Für die Positionsbestimmung wird die Höhe des Text von Textanfang bis Selektionsindex gespeichert. Ausgehend von dem Selektionsindex wird der Index nun jeweils um die Länge des letzten Wortes verkürzt. Sobald dadurch die Höhe des Textes verringert wird – der Text also eine Zeile weniger benötigt –, kann die Länge der letzten Zeile aus der Größe der Box um den Text zwischen Selektionsindex und Index vor der Verringerung berechnet werden (siehe Abbildung 6.5). Aus der Höhe des Textblocks kann die Y-Koordinate berechnet werden. Diese Methode funktioniert in den meisten Fällen, ist aber von der Arbeitsweise der Layoutengine abhängig, wie diese den Text umbricht. Sollte sich daran etwas ändern, funktioniert die Methode nicht mehr.

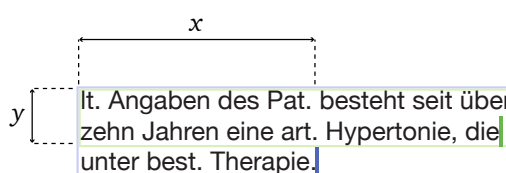


Abbildung 6.5: Berechnung der Koordinaten zum Einfügen des Textes. Der blaue Rahmen stellt die Begrenzung des Textes bis zur blauen Einfügemarke dar. Wird die Größe des Textes bis zur grünen Marke berechnet, so entsteht das grüne Rechteck.

6.3 Controller

Controller im iOS-SDK sind in einen bestimmten Darstellungsbereich verantwortlich für die Koordination von Interface-Elementen, die Reaktion auf Benutzerinteraktionen und das Übersetzen der Informationen vom Model in die Benutzeroberfläche. Alle Controller sind abgeleitet von der Klasse `UIViewController` und sind immer einem View zugeordnet.

Sie sind in die sogenannte *responder chain* (etwa „Antwortsender-Kette“) eingebettet. Alle Views, Windows und die Applikation selbst befinden sich in dieser Kette. Wenn der Benutzer mit dem Finger den Bildschirm berührt, wird ein Ereignis erzeugt. Angefangen vom tiefsten Element in der *responder chain* wird jeweils erfragt, ob das View-Objekt das Ereignis verarbeiten möchte. Wenn nicht, wird das Ereignis weiter die Ereigniskette heraufgereicht. Eine Ausnahme sind Views, welche einem Controller zugeordnet sind. Diese Views geben stattdessen das Ereignis an ihren Controller weiter. Hier kann der Controller entscheiden, was mit diesem Event passieren soll. Controller werden außerdem über Orientierungsänderungen informiert.

View Controller können andere modal anzeigen oder sie als Kinderelemente unterordnen und verwalten. Es gibt verschiedene Unterklassen von View Controllern, die spezielle User Interface Patterns unterstützen. Beispielsweise `UINavigationController`: Dieser Controller arbeitet wie ein Stack. Intial liegt ein View Controller

darauf. Jeweils der oben liegende Controller wird angezeigt. Weitere können auf den Stack geschoben werden (push). Beim Entfernen eines Controllers (pop) wird der vorherige angezeigt. Dem Benutzer kann eine Navigationsleiste präsentiert werden, die automatisch einen „Zurück“-Knopf enthält, sowie den Titel des aktuellen Controllers. Veränderungen werden entsprechend durch Animationen nach links (Hinzufügen) und rechts (Entfernen) animiert.

Controller können in Xcode grafisch orchestriert werden. Diese Modellierung wird auf einem sogenannten Storyboard vorgenommen. Durch Verbindungen zwischen Steuerelementen und Controllern kann der Programmierer angeben, dass durch eine Berührung dieser Controller eingeblendet wird. So kann sehr viel der Interaktion zwischen Controllern grafisch zusammengefasst und übersichtlich an einem Ort geschehen.

Abbildung 6.6 zeigt einen Ausschnitt aus dem Storyboard für *MEDo*. Zu sehen ist links der initiale Controller, der nach Applikationsstart angezeigt wird – ein Navigationscontroller (1). Dieser zeigt eine Login-Ansicht (2) an, auf den nach erfolgreicher Anmeldung der Stationsschirm (3) folgt. Von hier aus kann der Benutzer zum Patientenbildschirm (4) navigieren, eine neue Aufgabe erstellen (5) oder eine bestehende Aufgabe in einem modalen Fenster öffnen (6). Alle diese Controller werden im weiteren Verlauf dieses Abschnittes beschrieben und erklärt.

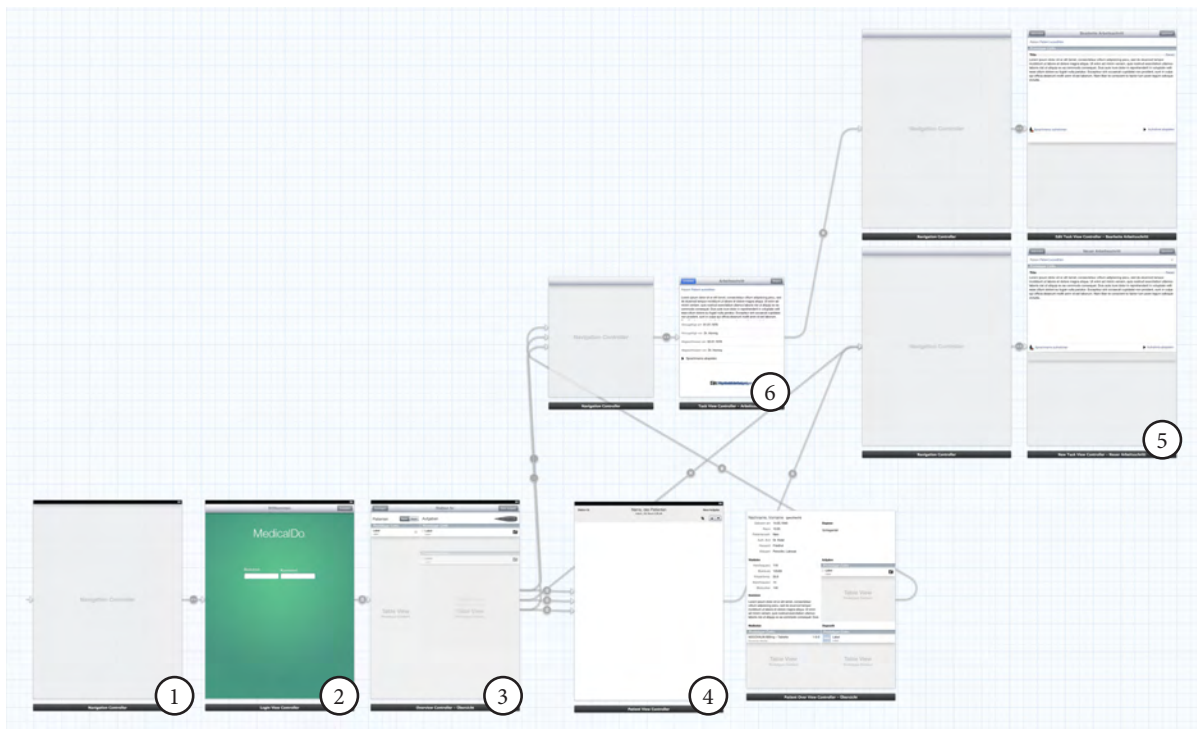


Abbildung 6.6: Storyboardausschnitt. Zu sehen sind (1) Navigationscontroller, (2) Login, (3) Stationsschirm, (4) Patientenbildschirm, (5) Neue Aufgabe und (6) bestehende Aufgabe anzeigen.

6.3.1 LoginViewController

Nach dem Start der Anwendung *MEDo* muss der Benutzer sich authentifizieren. Dies geschieht durch eine Kombination von Benutzernamen und Kennwort. Beides wird in zwei Textfelder eingetragen, die automatisch den Fokus bekommen (siehe Abbildung 6.8 auf der nächsten Seite). So kann der Benutzer direkt seinen Benutzernamen eintippen. Das System blendet automatisch eine Liste von zur aktuellen Eingabe passenden Benutzernamen ein, so dass der Name nicht komplett ausgetippt werden muss. Bei Auswahl eines Eintrags aus der Liste springt der Fokus in das Kennwortfeld. Nach dem dieses eingegeben wurde, tippt der Benutzer auf „Anmelden“ (oder „Öffnen“ auf der Tastatur).

Visuell ist der Anmeldebildschirm schlicht gehalten. Der Bildschirm wird von dem *MEDo*-Logo sowie dem Formular dominiert. Der grüne Hintergrund entspricht den Systemfarben. Der gleiche Hintergrund wird auch genutzt, um den kurzen Augenblick zu überbrücken während die Anwendung nach dem Start geladen wird (siehe Abbildung 6.7).

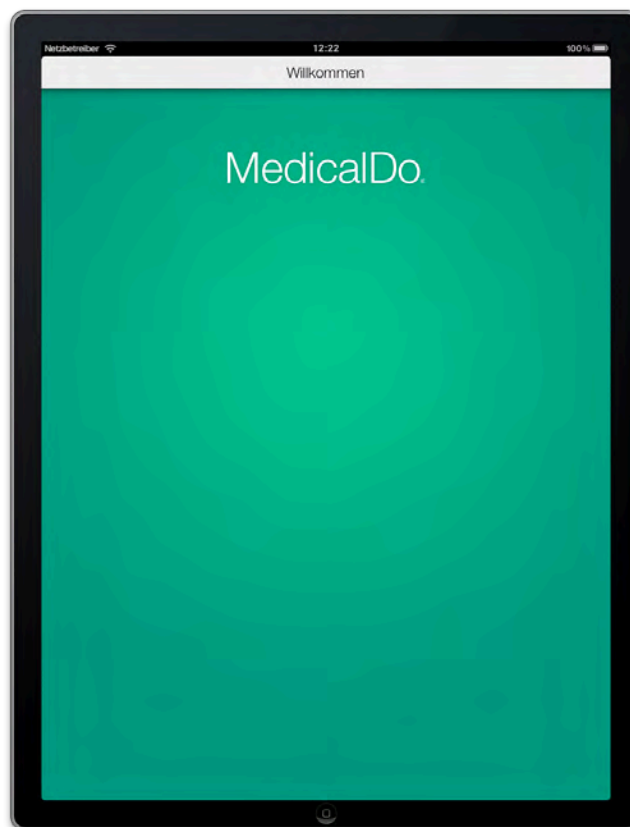


Abbildung 6.7: Start-Bildschirm

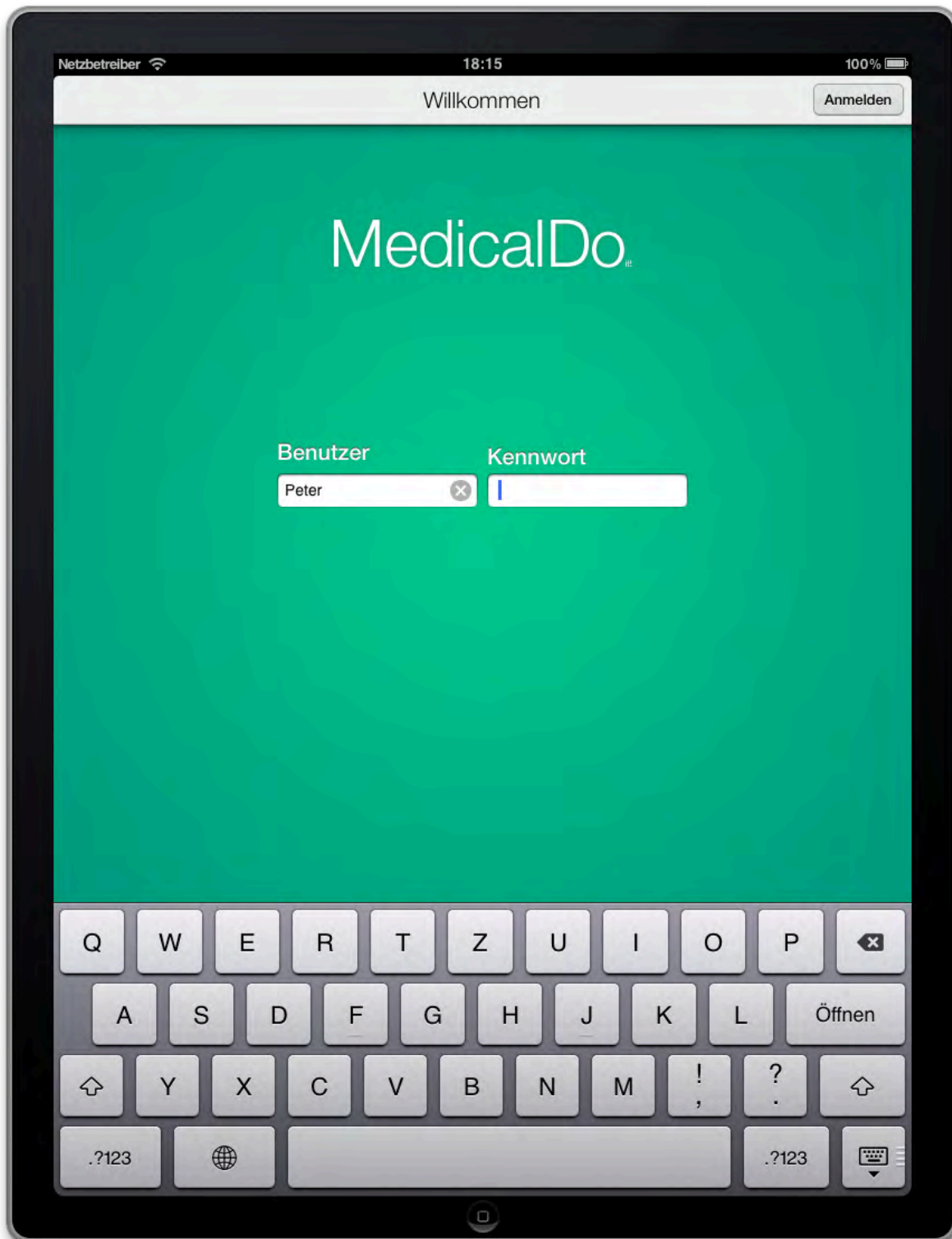


Abbildung 6.8: Login-Bildschirm

6.3.2 OverViewController

Die Stationsübersicht ist die zentrale Anlaufstelle für den Benutzer. Von hier aus können sowohl alle Patienten als auch alle Aufgaben aller Patienten eingesehen werden (siehe Abbildung 6.9 auf der nächsten Seite).

Der Stationsbildschirm enthält oben eine große Navigationsleiste, die den Titel des Bildschirms und zwei Buttons, die den Benutzer abmelden bzw. eine neue Aufgabe erstellen, enthält. Darunter befinden sich zwei Listen. Eine enthält alle Patienten, sortierbar nach Name oder nach Raum, die andere alle Aufgaben. Sind die Patienten nach Raum sortiert, werden Unterteilungen eingeblendet.

Die Aufgabenliste ist untergliedert in Bereiche für aktive und abgeschlossene Aufgaben und stellt im Normalfall die Aufgaben aller Patienten dar. Der Status wird zusätzlich über ein Markierungsfeld vor dem Text der Aufgabe angezeigt. Der Name des Patienten wird unter dem Aufgabentext eingeblendet. Neben der Aufgabe können verschiedene Symbole dargestellt werden: So kann angezeigt werden, dass es sich um einen Prozess handelt, der gerade bei einer anderen Person liegt (Zahnräder, graue Schrift), dass ein Formular auszufüllen ist (Dokument) oder Untersuchungsergebnisse anzuschauen sind (Röntgenbild). Es gibt die Möglichkeit, die Aufgabenliste nur nach den Aufgaben für einen bestimmten Patienten zu filtern, indem der kleine Pfeil neben dem Namen des Patienten angetippt wird.

Jede dieser Tabellen ist an eine Datenquelle angeschlossen, die die Daten aus dem Model in die Benutzeroberfläche übertragen. An dem Beispiel der Patientenliste soll die Funktionsweise dieses Objekts erklärt werden. Eine Datenquelle für eine Tabelle muss das Protokoll `UITableViewDataSource` implementieren. Es definiert Endpunkte, die die Tabelle anfragen kann, um sich selbst darzustellen.

Die wichtigsten Methoden sind `tableView:NumberOfRowsInSection:` und `tableView:cellForRowAtIndexPath:`. Die erste Funktion macht der Tabelle bekannt, wieviele Zellen diese anzeigen muss. Listing 1 auf Seite 65 demonstriert den Ablauf der zweiten Funktion. Die aufrufende Tabelle erwartet von dieser Methode eine Tabellenzelle (`UITableViewCell`) zurück, die auf die Tabelle gezeichnet wird.

Diese Tabellenzellen sollten aus Performanzgründen wiederverwendet werden. Das heißt, dass Tabellenzellen, die aus dem dargestellten Bild verschwinden, nicht vernichtet, sondern auf einen Stack gelegt werden. Sobald eine neue Tabellenzelle – beispielsweise durch Scrollen – dargestellt werden muss, wird keine neue erstellt, sondern stattdessen eine bereits existierende vom Stack genommen. So kann der durch Löschen und Erstellen von Objekten im Speicher entstehende Overhead vermieden werden.

Der richtige Patient wird durch den Parameter `indexPath` ermittelt. Dieser zweiteilige Index enthält jeweils einen Wert für die Sektion (Tabellen können in Sektionen unterteilt werden) und einen für die Tabellenreihe. Anhand dieser Daten und einem entsprechend aufgebauten, zweidimensionalen Array kann der passende Patient ausgewählt werden. Der Name wird anschließend in einem Label der Tabellenzelle darg-

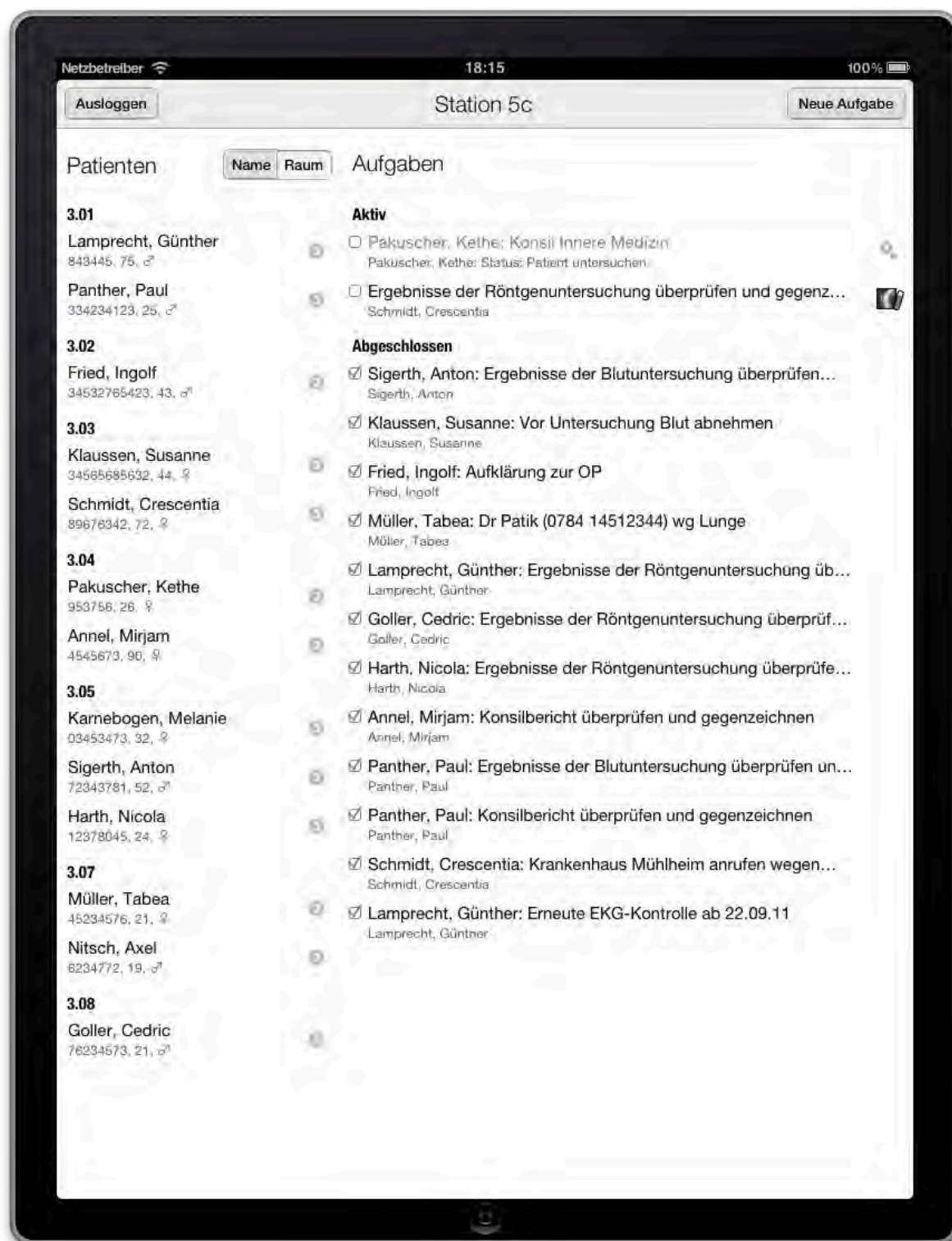


Abbildung 6.9: Stationsübersicht


```

- (UITableViewCell *)tableView:(UITableView *)tableView
  cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    // get the correct patient for the index path
    Patient *patient = [self patientForIndexPath:indexPath
                      inTableView:tableView];

    // dequeue one cell from the stack of unused cells
    static NSString *cellIdentifier = @"PatientCell";
    UITableViewCell *cell = [tableView
        dequeueReusableCellWithIdentifier:cellIdentifier];

    // [...]
    // fill the cell with our patients data
    cell.textLabel.text = [NSString stringWithFormat:@"%s, %s",
        patient.lastName, patient.firstName];

    // [...]
    return cell;
}

```

Listing 1: Befüllen einer Tabellenzeile mit Inhalt

stellt. Auf diese Weise aufgebaute Datenquellen können einfach an anderer Stelle im Projekt wiederverwendet werden.

6.3.3 PatientViewController

Die Patientenansicht besteht aus mehreren Unterkomponenten. Der Bildschirm ist unterteilt in den oberen Navigationsbereich, einer Reiterleiste sowie den Inhalten der Reiter darunter (siehe Abbildung 6.10 auf der nächsten Seite).

Der Navigationsbereich am oberen Bildschirmrand ist größer als die normale Navigationsleiste in iOS. Nur so passen alle benötigten Buttons in die begrenzte Fläche. Zusätzlich können so Geschlecht, Alter und Bett des Patienten angezeigt werden. Dies sind die wichtigsten Informationen zum Patienten und sind auf den anderen, den Patienten betreffenden, Bildschirmen auch verfügbar. Wie auf der Stationsübersicht kann auch hier eine neue Aufgabe durch einen Button oben rechts erstellt werden. Diese gleiche Position macht es für den Benutzer leichter die Funktion wiederzufinden, egal auf welchem Bildschirm er sich befindet. Buttons zum direkten Vor- und Zurücknavigieren zwischen Patienten in alphabetischer Reihenfolge werden ebenfalls angeboten. Wichtiger ist aber der Button, um direkt die Patientenliste anzuzeigen. Hier hat der Benutzer die gleichen Sortieroptionen wie auf dem Startbildschirm, ohne dass er seinen aktuellen Kontext verlassen muss.

Die Reiter unter dem Navigationsbereich teilen die Patientendaten logisch in vier Kategorien (exklusive der Übersicht), entsprechend Abschnitt 4.3 auf Seite 34: Anamnese, Vitaldaten, Diagnostik und Medikation. Diese werden in den folgenden Abschnitten erläutert.

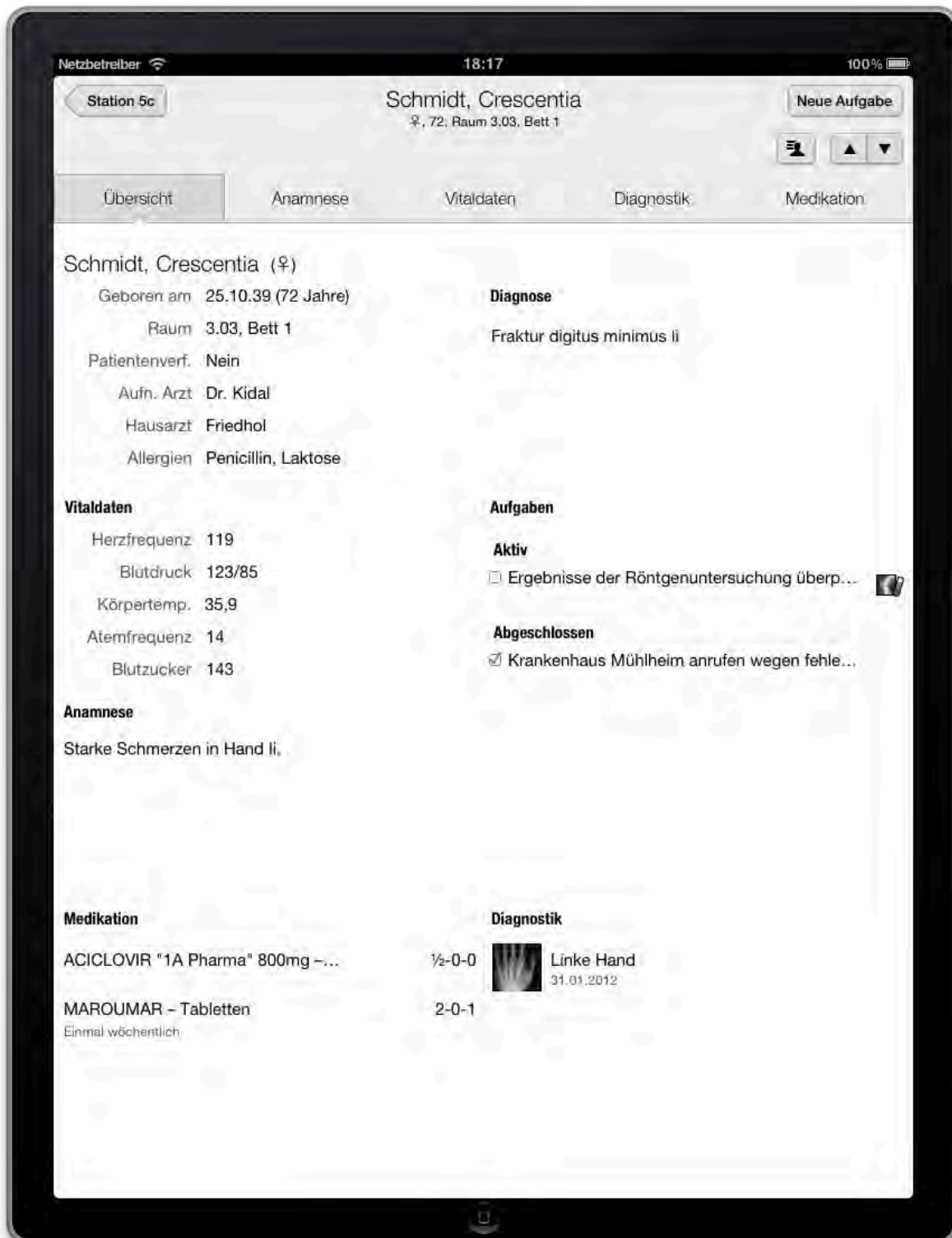


Abbildung 6.10: Patientenbildschirm

Übersicht

Die Übersicht fasst alle Daten, die meist auf den anderen Reitern zu finden sind, gesammelt in einer Sicht zusammen. So können Anwender sich schnell einen Überblick über den Status und den Fortschritt des Patienten machen. Es werden allgemeine Informationen wie das Alter, in welchem Bett der Patient liegt, ob er über eine Patientenverfügung verfügt, welcher Mediziner ihn aufgenommen hat, Kontaktinformationen und Name des Hausarztes sowie eventuelle Allergien angezeigt. Wichtig ist, dass Diagnose und Anamnese schnell einsehbar sind [53]. Der Anwender hat von hier aus auch Zugriff auf die Aufgaben.

Diagnostika

Bei der Visite sind die verschiedenen Diagnostika die essentiellen Daten. Diese können in diesem Reiter eingesehen werden. Alle Untersuchungsergebnisse des Patienten werden in Listenform auf der linken Seite angezeigt, während Details zur ausgewählten Untersuchung rechts im größeren Bereich dargestellt werden (siehe Abbildung 6.11 auf der nächsten Seite). Abhängig vom Untersuchungstyp wird im rechten Bildschirmbereich einer der drei Controller `DiagnosticImageViewController` (Ergebnisse von bildgebenden Untersuchungen), `CouncilViewController` (Konsile) und `TestResultViewController` (Laborergebnisse) instanziiert.

Bildgebende Untersuchungen Der `DiagnosticImageViewController` baut auf dem `Photo Album View Controller` des `Nimbus Framework`¹ auf und stellt eine Datenquelle für diesen Controller bereit. Diese Datenquelle ist ähnlich der von Tabellen aufgebaut, liefert allerdings Bilder statt Tabellenzellen zurück. Die Komponente aus dem `Nimbus Framework` wird von einem Anzeigebereich dominiert, den der Controller mit einem Bild füllt. Durch Wischen nach links und rechts kann zwischen Bildern horizontal navigiert werden. Zoomen und Verschieben des Bildausschnitts mittels Gesten werden auch unterstützt. Zur Übersicht wird außerdem eine Werkzeugleiste im unteren Bildschirmbereich eingeblendet, die Thumbnails der Bilder enthält.

Wichtig bei der Implementierung war in Anbetracht der meist hochauflösten Röntgenbilder, Caching für die Thumbnails und Nebenläufigkeit zu realisieren. Ansonsten kann es zu Blockierungen der Benutzerschnittstelle kommen.

iOS bietet verschiedene Arten um mit Nebenläufigkeit umzugehen. Bei der Wahl sollte mit der Methode mit der größten Abstraktion begonnen werden. Je nach Bedarf sollte dann auf kompliziertere Methoden zurückgegriffen werden. Im konkreten Fall wurde `Grand Central Dispatch` (GCD) eingesetzt. GCD macht es einfach für mehrere Prozessoren zu programmieren, indem es von diesen abstrahiert. Die Vorgehensweise, um mithilfe von GCD Vorschaubilder zu konstruieren, wird in Listing 2

¹<http://nimbuskit.info/>, Abruf am 15. März 2012



Abbildung 6.11: Patientenbildschirm – Diagnostika

vorgestellt.

```
// obtain queue and add task to it:
// loading all images and storing a thumbnail of them in an array
dispatch_queue_t queue = dispatch_get_global_queue(
    DISPATCH_QUEUE_PRIORITY_LOW, 0ul);
dispatch_async(queue, ^{
    NSMutableArray * thumbnails = [NSMutableArray
        arrayWithCapacity:self.diagnosticImages.count];
    for (DiagnosticImage *diagnosticImage in self.diagnosticImages)
    {
        // generate thumbnail and add to thumbnail array[...]
    }
    // add thumbnails to toolbar (in main/gui thread)
    dispatch_sync(dispatch_get_main_queue(), ^{
        // [...]
    });
});
```

Listing 2: Einsatz von GCD um Vorschaubilder zu generieren

Wichtig ist dabei, dass alle Operationen, die die GUI betreffen, im Hauptthread ausgeführt werden. Analog zu diesem Code geschieht auch das Laden der großen Bilder. Diese werden asynchron aus der Datenbank geladen, da die Animationen ansonsten nicht flüssig angezeigt werden.

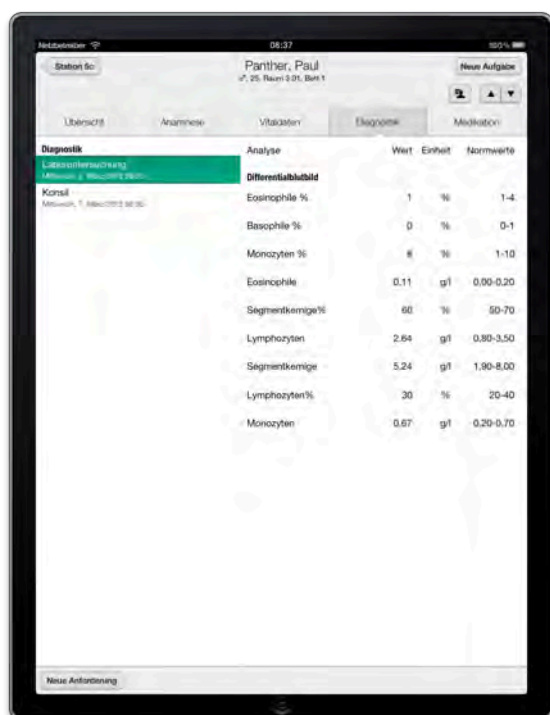
Konsile Der CouncilViewController ist sehr einfach aufgebaut, da hier nur der Text und die Überschrift des Konsils dargestellt werden muss.

Laboreergebnisse Ergebnisse aus dem Labor werden in einer einfachen Tabellen-Ansicht dargestellt (siehe Abbildung 6.12 auf der nächsten Seite). Diese ist in Sektionen, entsprechend den Teiluntersuchungen geordnet. Die Tabelle bleibt durch eine feste, nicht mitscrollende Tabellenkopfzeile lesbar, wenn der Benutzer weit nach unten gescrollt hat. Der Controller ist gleichzeitig auch Datenquelle für die Tabelle. Wenn neue Testergebnisse gesetzt werden, bringt er diese auch in eine für die Darstellung zweckdienlichere Datenstruktur.

Untersuchungen anfordern Der Benutzer kann direkt aus dem Diagnostikbildschirm weitere Untersuchungen anfordern. Anforderungen werden in der Liste der Diagnostika in einer eigenen Sektion angezeigt. Hier verbleiben diese, bis sie abgeschickt werden. Die Anforderungen sind als Datentyp Form modelliert. Alle Formulare bieten unten rechts eine Option, um sie abzusenden. Das System kennt drei verschiedene Formulare, die den unterstützten Prozessen entsprechen: Konsil, Röntgen- und Laboruntersuchung. Dabei sind sich Konsil- und Röntgenanforderung insofern ähnlich, dass bei beiden die Eingabe durch Ausfüllen von Textfeldern geschieht:

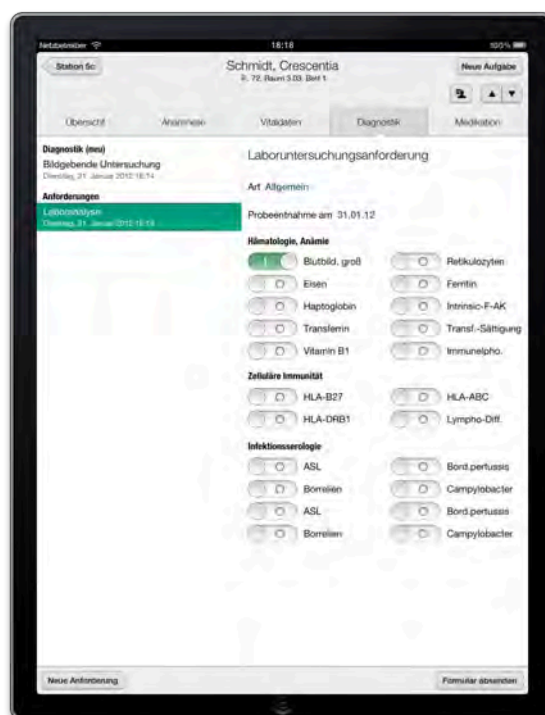
Anamnese, Indikation, Fragestellung und die gewünschte Untersuchung (nur bei Röntgenuntersuchungen).

Eine andere Darstellung wird bei der Anforderung einer Laboruntersuchung genutzt (siehe Abbildung 6.13). Hier werden dem Benutzer eine Liste von Schaltern angeboten, durch die er die gewünschten Untersuchungen aktiviert. Um dies für den Anwender zu vereinfachen, gibt es zusätzlich Schalter, die häufig genutzte Untersuchungen zusammenfassen (wie große und kleine Blutuntersuchungen). Diese Modellierung ist analog zu den Laboranforderungen gehalten, wie sie in den Krankenhäusern eingesetzt werden.



Analyse	Wert	Einheit	Normwerte
Differentialblutbild			
Eosinophilie %	1	%	1-4
Basophilie %	0	%	0-1
Monozyten %	8	%	1-10
Eosinophilie	0.11	g/l	0.00-0.20
Segmentkernige	60	%	50-70
Lymphozyten	2.64	g/l	0.80-3.50
Segmentkernige	5.24	g/l	1.90-8.00
Lymphozyten%	30	%	20-40
Monozyten	0.67	g/l	0.20-0.70

Abbildung 6.12: Patientenbildschirm – Laborergebnisse



Laboruntersuchungsanforderung

Art Allgemein:

Probeentnahme am 31.01.12

Hämatologie, Anämie

☒ Blutbild, groß ☐ Retikulozyten

☐ Eisen ☐ Ferritin

☐ Haptoglobin ☐ Intrinsic-F-AK

☐ Transferrin ☐ Transf.-Sättigung

☐ Vitamin B1 ☐ Immunelepho.

Zelluläre Immunität

☐ HLA-B27 ☐ HLA-ABC

☐ HLA-DRB1 ☐ Lympho-Diff.

Infektionsurologie

☐ ASL ☐ Bord pertussis

☐ Borrelien ☐ Campylobacter

☐ ASL ☐ Bord pertussis

☐ Borrelien ☐ Campylobacter

Abbildung 6.13: Patientenbildschirm – Laboruntersuchung anfordern

Medikation

Die Medikation ist im Demonstrator statisch. Es werden zwei Tabellenzeilen eingeblendet, die demonstrieren sollen, wie der Bildschirm aussehen könnte (siehe Abbildung 6.14 auf der nächsten Seite).

Vitaldaten

Genauso wie bei der Medikation handelt es sich bei den Vitaldaten um einen statischen Bildschirm, um das System glaubhaft zu gestalten (siehe Abbildung 6.15 auf der nächsten Seite).



Abbildung 6.14: Patientenbildschirm – Medikation



Abbildung 6.15: Patientenbildschirm – Vitaldaten

6.3.4 NewTaskViewController

Neue Aufgaben können schnell über den immer erreichbaren „Neue Aufgabe“-Button auf jedem Screen in der oberen rechten Ecke erstellt werden. Der `NewTaskViewController` ist für die Annahme dieser Informationen konzipiert. Er bietet eine Patientenauswahl, ein Texteingabefeld, ein Modul um Textbausteine einzufügen und einen Button um eine Audioaufnahme zu starten (siehe Abbildung 6.16 auf der nächsten Seite). Dem Controller zugrunde liegt eine Tabelle, bei der die einzelnen Elemente in den verschiedenen Zellen liegen.

Die Patientenauswahl ist so gestaltet, dass nach Antippen darunter eine Liste von Patientennamen ausklappt. Anfangs wurde hier ein Popup eingesetzt, welches allerdings eine weitere Ebene einführt, die Interaktionen mit anderen Elementen der Sicht nicht ermöglicht. In dieser Liste kann der Benutzer durch Wischen schnell zum gewünschten Patienten scrollen, ohne dass dadurch die Interaktion mit den anderen Elementen auf der Seite gestört ist. Die Patientenliste wird ausgeblendet, sobald ein Patient ausgewählt wird.

Eingaben auf der Bildschirmtastatur, die automatisch eingeblendet wird, werden in das Textfeld zur Eingabe des Aufgabentextes gerichtet. Zusätzlich kann der Benutzer durch die Textbausteine darunter schnell häufig benutzte Ausdrücke einfügen. Abschnitt 6.2.1 auf Seite 57 erklärt die Implementierung der Komponente.

Neben den beiden genannten Optionen gibt es noch die Möglichkeit, weitere In-

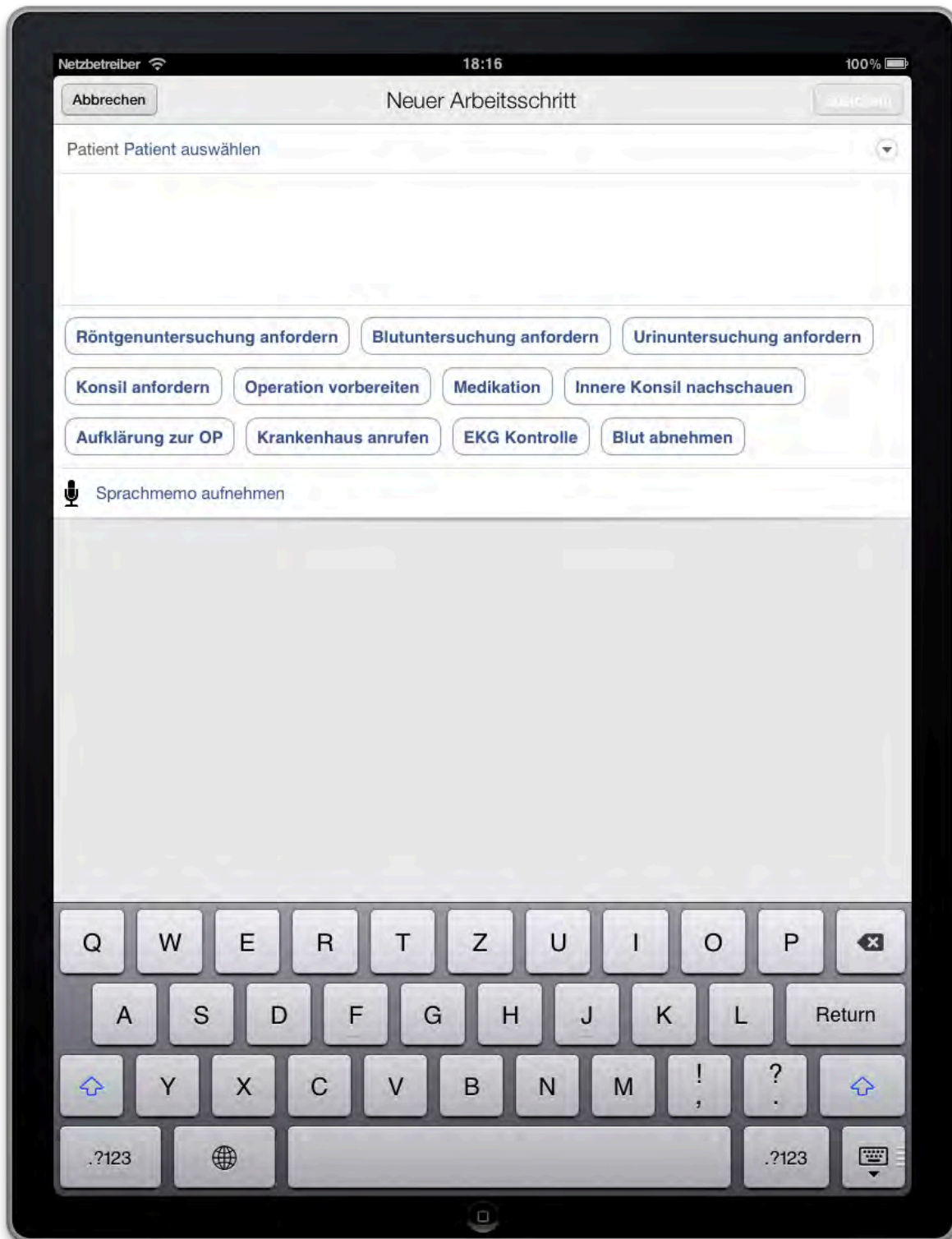


Abbildung 6.16: Bildschirm: Neue Aufgabe erstellen

formationen per Audioaufnahme der Aufgabe hinzuzufügen. Dafür wird intern eine Instanz der Klasse `AVAudioRecorder` aus dem `AVFoundation` Framework genutzt. Sie bietet eine Abstraktion, mit der sehr einfach Audio aufgenommen werden kann. Startet der Benutzer den Aufnahmevorgang, beginnt der Audiorekorder, die Daten in eine temporäre Datei zu schreiben. Beendet der Benutzer die Aufnahme, lädt der Controller die Daten in den Speicher. So kann der Benutzer sich direkt die Aufnahme anhören, wofür wiederum eine Instanz von `AVAudioPlayer` genutzt wird. Damit der Benutzer ein visuelles Feedback erhält, dass die Aufnahme funktioniert und er später auch zu verstehen ist, wird eine kleine Anzeige eingeblendet, die die Aufnahmelautstärke durch die Helligkeit eines Kreises zeigt. Um die Darstellung zu skalieren, wird die aktuelle Lautstärke durch die Durchschnittslautstärke geteilt.

6.3.5 TaskViewController

Um dem Benutzer die Orientierung zu erleichtern, ähneln sich die Ansichten zum Anzeigen und zum Erstellen von Aufgaben stark. Ausser dem Patientennamen werden Aufgabentext, Erstellungsdatum und Autor der Aufgabe angezeigt. Durch Berühren des Patientennamens kann der Benutzer schnell zur Detailansicht des Patienten springen. Bei abgeschlossenen Aufgaben wird zusätzlich angezeigt, wer wann die Aufgabe abgeschlossen hat. Hat der Ersteller der Aufgabe eine Audioaufnahme erstellt, so wird ein Button angezeigt, um die Aufnahme wiederzugeben und zu stoppen.

Ein großer Button unter dieser Anzeige zeigt die zur Aufgabe passenden Aktionen an. Wenn beispielsweise ein Röntgenbild eingetroffen ist, leitet der Button den Benutzer zum entsprechenden Bild. Bei auszufüllenden Formularen wird zum Formular navigiert. Handelt es sich um eine einfache Aufgabe, ohne angebundenen Prozess, markiert dieser Button die Aufgabe als erledigt und blendet das Fenster aus.

Der Benutzer hat jederzeit die Möglichkeit, von ihm selbst angelegte Arbeitsschritte zu ändern. Hierzu braucht er nur auf den „Ändern“-Button in der rechten oberen Ecke des Fensters zu tippen. Es öffnet sich ein von `NewTaskViewController` erben- des Fenster, bei dem die Auswahl des Patienten deaktiviert ist. So können bereits erstellte Aufgaben schnell geändert werden.

6.3.6 MDPortraitSplitViewController

Ein häufiges Muster sind Ansichten, die gleichzeitig eine Übersicht anbieten (*Master*), auf Wunsch aber auch Details zu einem Punkt anzeigen (*Detail*). Als Beispiel kann eine E-Mail-Anwendung betrachtet werden. Im linken Bereich werden alle E-Mails im aktuellen Postfach angezeigt. Bei Auswahl einer E-Mail wird diese daraufhin im rechten Bereich angezeigt. Eine Variation ist die Anzeige über- beziehungsweise untereinander. Dieses Designmuster wird von Tidwell [54, S. 31] *two panel*

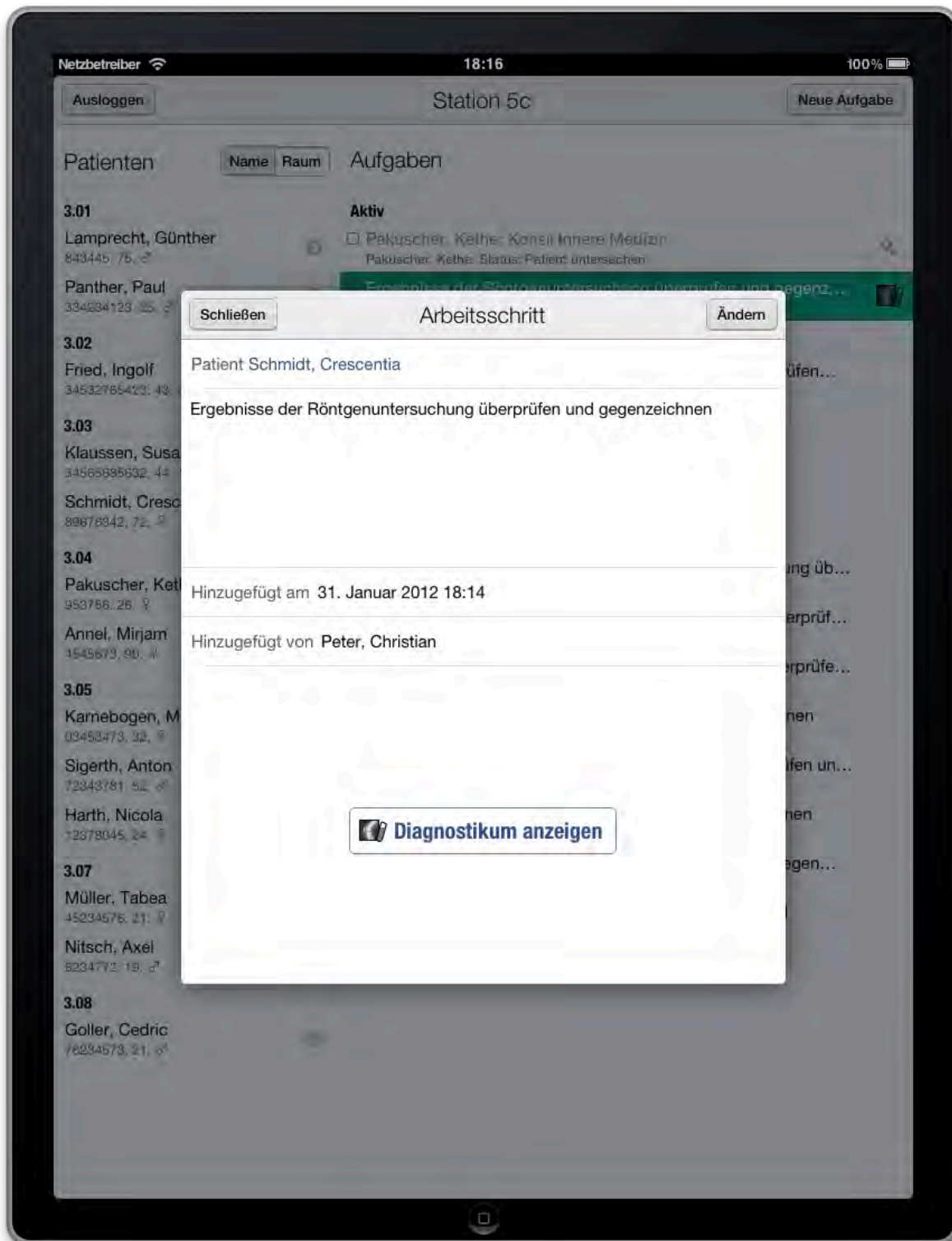


Abbildung 6.17: Bildschirm: Aufgabendetails anzeigen

*selector*² genannt. Auf dem iPad wird es von Apple durch einen speziellen Controller unterstützt, den sogenannten `SplitViewController`.

Da ein iPad verschiedene Orientierungen unterstützt und die Portraitorientierung dabei horizontal weniger breit ist, wird in dieser Orientierung der *two panel selector* aufgegeben. Stattdessen wird ein Button in der Navigationsleiste platziert, mithilfe dessen der Benutzer die *Master*-Liste anzeigen kann (siehe dazu Abbildung 6.18). So kann horizontal Platz eingespart werden.

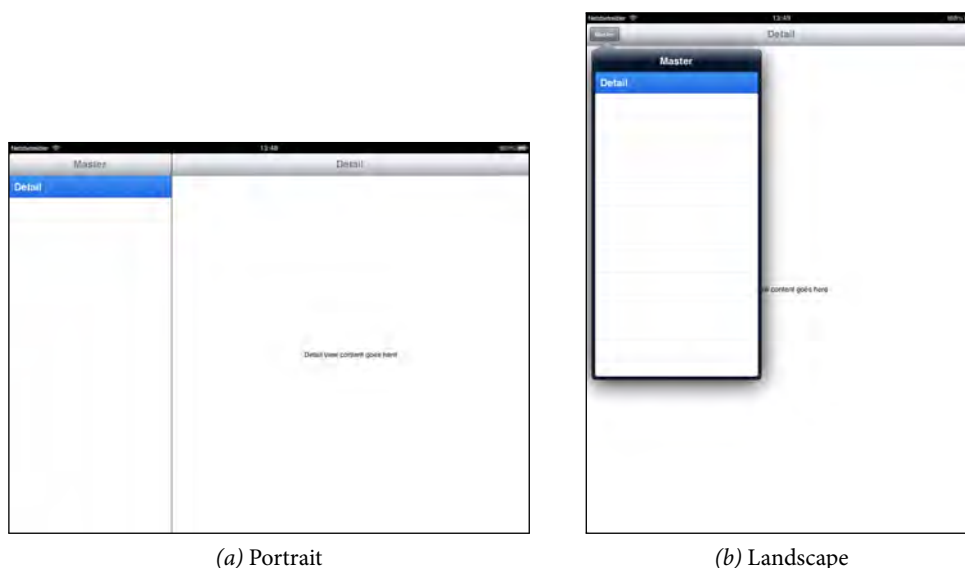


Abbildung 6.18: Darstellung des normalen `UISplitView` in verschiedenen Orientierungen

Für die Übersicht der Diagnostika ist dies nicht erwünscht, da einerseits die Auswahl über den Button umständlich ist (und nicht sehr einleuchtend für den Benutzer) und andererseits der Button bedeuten würde, unter der Patientenleiste eine weitere Toolbar einzusetzen. Es musste also eine eigene Klasse geschrieben werden, die diese Split-Funktionalität besitzt und auch im Portrait-Modus anzeigt. Zusätzlich war aber auch eine Anforderung, dass der Benutzer die Master-Ansicht ausblenden kann, falls er mehr vom Detailinhalt sehen möchte. Die wichtigsten Attribute, die diese Klasse also öffentlich machen sollte, sind die Controller für Detail- und Masteransicht. Entsprechend einfach ist das Interface von `MDPortraitSplitViewController`:

```
@interface MDPortraitSplitViewController : UIViewController
@property (strong, nonatomic) UIViewController *masterViewController;
@property (strong, nonatomic) UIViewController *detailViewController;
@end
```

Jeweils wenn der Master- oder der Detail-Controller gesetzt werden, wird das entsprechende View auf der controllereigenen View platziert und der Controller

²Mehr Informationen zum *two-panel selector* Pattern finden sich auf der Website der Autorin: <http://designinginterfaces.com/patterns/two-panel-selector/>

als Child-Controller hinzugefügt. Dem Master-Controller wird zusätzlich noch ein Swipe Gesture Recognizer hinzugefügt. Gesture Recognizer erkennen Gesten wie Swipes (Wischgesten), Pinch (Finger zusammenziehen) usw. Dazu stehen verschiedene Unterklassen der abstrakten Klasse `UIGestureRecognizer` zur Verfügung, die jeweils eine Gestenart erkennen. Der Gesture Recognizer im `MDPortraitSplitViewController` soll Wischgesten nach links erkennen (`UISwipeGestureRecognizer` mit dem Attribut `direction = UISwipeGestureRecognizerDirectionLeft`). Auf diese Weise kann der Benutzer die Master-Liste auf der linken Seite durch ein Wischen nach links animiert verkleinern. Da es sich um eine kleine Bewegung handelt, reicht eine Dauer von ca. 0,2 Sekunden. Zusätzlich zur Bewegung wird auch die Liste verdunkelt, so dass diese inaktiv erscheint.

Um die Master-View wieder anzuzeigen, kann der Benutzer entweder wieder eine Wischbewegung nach rechts machen oder die Liste berühren.

6.4 Workflowunterstützung im System

Um den Arbeitsaufwand im Rahmen zu halten, und weil es keinen Einfluss auf die Wahrnehmung beim Benutzer hat, wurde auf die Unterstützung oder Anbindung eines richtigen WfMS verzichtet. Stattdessen wurde eine sehr einfache, automatische Unterstützung eingebaut, die von einem Status zum nächsten weiterwechselt, dabei aber keine Verzweigungen oder ähnliches bietet. Das System basiert auf einer XML-Datei, die alle unterstützten Prozesse enthält. Diese Prozesse besitzen eine ID, einen menschenlesbaren Titel, Stichworte, sowie eine Liste von Arbeitsschritten, die ausgeführt werden müssen, um den Prozess zu bearbeiten.

6.4.1 XML-Datei

Das Listing 3 auf der nächsten Seite zeigt einen Ausschnitt aus der XML-Datei, die für die Applikation benutzt wird. Wenn eines der Stichworte (Phrase) in einer neuen Arbeitsaufgabe gefunden wird, so nimmt das System automatisch an, dass der Benutzer einen solchen Prozess starten möchte. Als Beispiel: Gibt der Benutzer im Aufgabentext „Blutanalyse“ ein, so wird diese Aufgabe automatisch zum einen mit dem Prozess `BloodAnalysis` verknüpft, zum anderen wird ein Formular für die Laboranforderung erstellt.

Beim Absenden des Formulars schaltet das System zur ersten Aufgabe des Prozesses weiter. Im Falle der Blutuntersuchung wäre der nächste Schritt „Patient für Blutuntersuchung vorbereiten, Proberöhrchen bereitlegen“. Dieser Schritt soll von der Rolle `nurse` ausgeführt werden, weswegen der entsprechende Wert im Attribut `role` des Arbeitsschritts gesetzt ist. Außerdem besitzen Arbeitsschritte noch ein `ID`-Attribut sowie den Text der Aufgabe als Wert. Die Pflegekräfte müssen nun manuell ihren Arbeitsschritt bestätigen, damit das System zum nächsten Schritt weiterschaltet.

```

<Processes>
  <Process id="BloodAnalysis" title="Blutuntersuchung">
    <Phrases>
      <Phrase>Laboranalyse</Phrase>
      <Phrase>Blutuntersuchung</Phrase>
      <Phrase>Blutanalyse</Phrase>
      <Phrase>Blutuntersuchung vervollständigen</Phrase>
    </Phrases>
    <Tasks>
      <Task id="1" role="nurse">
        Patient für Blutuntersuchung vorbereiten,
        Proberöhrchen bereitlegen
      </Task>
      <Task id="2" role="physician">
        Blut abnehmen
      </Task>
      <Task id="3" role="laboratory">
        Blut untersuchen
      </Task>
      <Task id="4" role="physician" action="labResult">
        Ergebnisse der Blutuntersuchung überprüfen
        und gegenzeichnen
      </Task>
    </Tasks>
  </Process>
  <Process id="XrayExamination" title="Röntgenuntersuchung">
    ...
  </Process>
  <Process id="Council" title="Konsil">
    ...
  </Process>
</Processes>

```

Listing 3: Definition der Prozesse in XML

tet. Auf diese Weise wird schließlich der gesamte Ablauf erfasst und kann so vom System sukzessive abgearbeitet werden.

Eine Besonderheit ist das Attribut `action`, welches angibt, dass dem Benutzer etwas präsentiert werden soll. Im Falle der Blutuntersuchung wäre das „labResult“. Das bedeutet, dass der Benutzer erst das Ergebnis anschauen und als gesehen markieren muss.

6.4.2 MDProcessManager

Im System ist für die Verwaltung der Prozessinstanzen die Klasse `MDProcessManager` verantwortlich. Diese hat folgendes Interface:

```

@interface MDProcessManager : NSObject
+ (Task *)advanceProcessInstance:(ProcessInstance *)processInstance
    advanceAutomatically:(BOOL)automatically
    withDelay:(BOOL)withDelay;
+ (Task *)advanceProcessInstance:(ProcessInstance *)processInstance;
+ (id)processInstanceWithTask:(Task *)task
    andPatient:(Patient *)patient;
@end

```

Neue Prozesse instanziiieren

Erstellt ein Benutzer eine neue Aufgabe, so wird im System zuerst ein neuer Task mit dem entsprechenden Text erstellt. Dieser wird an die Methode `processInstanceWithTask:andPatient:` weitergegeben, die einen Prozess zu der Aufgabe erstellt. Durch Abgleich des Textes mit den bereits beschriebenen Ausdrücken (Phrases) wird entschieden, ob der Aufgabe ein Prozessmodell zugewiesen wird. Als Interface wird der DOM-Parser TBXML³ genutzt. Das Listing 4 auf der nächsten Seite zeigt, wie mithilfe von TBXML alle Ausdrücke durchsucht werden.

Weiterschalten von Prozessen

Wenn ein Arbeitsschritt erledigt wurde, wird `advanceProcessInstance:` aufgerufen. Aus Demonstrationszwecken existiert auch die Methode `advanceProcessInstance:advanceAutomatically:withDelay:`. Hier wird der Prozess ebenso weitergeschaltet. Fordert die neue Aufgabe allerdings eine andere Rolle als die des aktuell angemeldeten Benutzers und ist das Flag `advanceAutomatically` gesetzt, so wird der Prozess automatisch weitergeschaltet. Das Attribut `withDelay` gibt dabei an, ob dies mit einer Verzögerung geschehen soll. Im Demonstrationssystem sind es 5 Sekunden. Es wird im Demobetrieb immer diese zweite Methode mit beiden Attributen auf YES ausgeführt. So werden nach und nach alle Prozessschritte durchgespielt und der Benutzer kann sehen, wie das System den Status anzeigt und aktualisiert.

Im Detail wird zuerst der aktuelle Prozess in der Prozessbeschreibung (XML-Datei) anhand der Prozess-ID gesucht. Innerhalb dieses wird nun der Prozessschritt anhand dessen Task-ID gesucht. Eine Ausnahme ist der Fall, wenn die aktuelle Aufgabe keine ID besitzt. Dies ist genau dann der Fall, wenn es sich um die erste Aufgabe in einem Prozess handelt, also die, mit der der Prozess erst angestoßen wird – wie das Absenden eines Formulars.

6.5 Zusammenfassung

Mithilfe des iOS-SDKs konnte sehr schnell ein lauffähiger Prototyp entwickelt werden, der die Aspekte Prozessmanagement, Arbeitsliste sowie Mobilität in einer Ap-

³<https://github.com/71squared/TBXML>, Abruf am 15. März 2012

```

TBXML *tbxml = [TBXML tbxmlWithURL:processesDescriptionURL];
// iterate through all processes
TBXMLElement *processElement =
    [TBXML childElementNamed:@"Process"
     parentElement:tbxml.rootXMLElement];
do
{
    NSString *processID = [TBXML valueOfAttributeNamed:@"id"
                              forElement:processElement];
    TBXMLElement *phrasesElement = [TBXML childElementNamed:@"Phrases"
                                         parentElement:processElement];
    TBXMLElement *phraseElement = [TBXML childElementNamed:@"Phrase"
                                         parentElement:phrasesElement];

    do
    {
        // check for phrase in task text
        NSString *phrase = [TBXML textForElement:phraseElement];
        if ([task.text rangeOfString:phrase
                                options:NSCaseInsensitiveSearch
                                ].location != NSNotFound)
        {
            foundPhrase = YES;
            break;
        }
        // iterate phrases
        phraseElement = phraseElement->nextSibling;
    } while (phraseElement);

    // iterate processes
    processElement = processElement->nextSibling;
} while (processElement);

// check if process id corresponds to one known to our system
[...]
```

Listing 4: Iteration durch die Ausdrücke der XML-Datei zur Erkennung des Prozesses

plikation vereint. Tabelle 6.1 auf der nächsten Seite vergleicht die in Abschnitt 4.1 auf Seite 30 erarbeiteten Anforderungen mit dem Prototyp der *MEDo*-Applikation. Dabei wurden alle Anforderungen erfüllt – bis auf die Forderungen nach einfacher Desinfizierung und nach Flexibilität der Prozesse. Bei ersterem ist ein iPad nicht zufriedenstellend, da es nicht komplett abgedichtet ist und deshalb nicht ausreichend desinfiziert werden kann. Flexibilität in Prozessen ist ein komplexes Thema und wäre eine eigene Arbeit wert. Aus diesem Grund konnte dies leider nicht in das System miteinfließen.

Tabelle 6.1: Abgleich der Anforderungen mit Implementierung

ANFORDERUNG	IMPLEMENTATIONSART	IMPLEMENTIERT
SYSTEM		
Abbildung eines papierbasierten Laufzettels	Primäre Eingabe bei Erstellung einer neuen Aufgabe ist Freitext; Einfache Aufgaben können als erledigt markiert werden.	✓
Einbeziehung von zusätzlichen Daten	Alle geforderten Daten (Aktivität, Zeitpunkt aufgenommen, Zeitpunkt bearbeitet, Benutzer aufgenommen, Benutzer bearbeitet, Zugeordnete Daten) werden aufgenommen.	✓
Arbeitslistenfunktion	Drei Prozesse (Laboruntersuchung, Röntgenuntersuchung und Konsil) werden vom System unterstützt.	✓
Unterbrechbarkeit	Formulare werden automatisch gespeichert; Es gibt verschiedene Möglichkeiten, Prozesse zu starten.	✓
Flexibilität	Flexibilität in Prozessen wird nicht unterstützt.	✗
Geschwindigkeit der Eingabe	Durch Textbausteine und die Möglichkeit, Audioaufnahmen an Aufgaben anzuhängen, können schnell Eingaben getätigt werden. Eine virtuelle Tastatur ermöglicht die Eingabe von Freitext.	✓
Rollen-/Mehrbenutzerunterstützung	Im System sind die Rollen Ärztin/Arzt, Pflegepersonal und eine Reihe von internen Untersuchungsstationen implementiert. Im System kann man sich nur als Arzt/Ärztin anmelden.	✓
Geteilte Listen	Alle Aufgaben sind von allen den Personen einsehbar, die die passende Rolle besitzen.	✓
Adäquate Anzeige der anstehenden Aufgaben	Dies wird durch die Übersicht auf dem Stationschirm und den Patientenschirmen gewährleistet.	✓
Adäquate Bearbeitung der Aufgaben	Wird ebenso unterstützt, durch entsprechende Buttons.	✓

Fortgesetzt auf der nächsten Seite

Tabelle 6.1 – fortgesetzt von der vorhergehenden Seite

ANFORDERUNG	IMPLEMENTATIONSART	IMPLEMENTIERT
Jederzeit Aufgabenbearbeitung abbrechen	Formulare werden automatisch gespeichert, Diagnostika erst nach aktiver Bestätigung als gesehen markiert.	✓
Unterstützung für Schichtübergabe	Indirekt, da jederzeit alle passenden Aufgaben angezeigt werden.	✓
Adäquate Anzeige von Untersuchungsergebnissen	Siehe Seite 65.	✓
Adäquate Benachrichtigung über Status von Untersuchungen	Aktive Untersuchungen werden angezeigt, Benachrichtigung über neue Untersuchungsergebnisse in der Aufgabenliste.	✓
Komplette Abdeckung der Möglichkeiten der Papier-Handzettel	Sollten mit den vorangegangenen Anforderungen abgedeckt sein.	✓
TABLET		
Größe	Als eines der größten Tablets mit 24,6 cm (9,7 Zoll) Bildschirmdiagonale ist das iPad gut aufgestellt.	✓
Blickwinkel	Mit 178° absolut ausreichend; Auch in mehrere Richtungen, da Gerät in mehreren Orientierungen genutzt werden kann.	✓
Hygiene	Nicht spritzwassergeschützt, nicht einfach desinfizierbar.	✗

Dieses Kapitel schilderte die Architektur der *MEDo*-Applikation, aufgeteilt in *Model*, *View* und *Controller*. Mit der iPad-Applikation konnte schließlich die Evaluation mit Ärzten durchgeführt werden, die im nächsten Kapitel erläutert wird.

7

Evaluierung

Mithilfe des entwickelten Prototypen sollte überprüft werden, ob ein System, das die Ärztinnen und Ärzte bei der Verwaltung ihrer Aufgaben unterstützt, von diesen angenommen wird. Weiterhin sollte die Usability der Applikation überprüft werden. Zu diesem Zweck wurde eine qualitative Umfrage sowie ein Usability-Test mit zehn Ärzten durchgeführt. Da bei dieser Evaluierung die Usability im Vordergrund steht, wurde eine qualitative Umfrage als hinreichend erachtet [5].

Dieses Kapitel stellt die Vorgehensweise und die Ergebnisse vor und schließt mit der Diskussion der Ergebnisse.

7.1 Ablauf

Für die Evaluation von *MEDo* wurde das Formblatt auf Seite 100 bis 102 genutzt. Nach der Begrüßung wurden der/dem Interviewpartner/in der Fokus der Arbeit erklärt. Es wurde darauf hingewiesen, dass es sich hier nur um einen Prototypen handelt, der nicht an die Daten der Station angeschlossen ist. Der gesamte Ablauf wird in Abbildung 7.1 dargestellt.

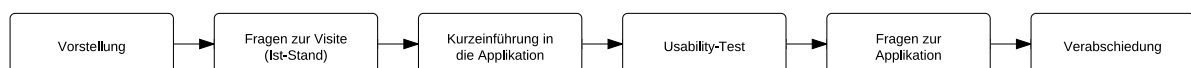


Abbildung 7.1: Ablauf des Evaluierungsprozesses

7.1.1 Erster Teil des Interviews

Um die Hintergründe der Ärzte besser verstehen zu können, war es wichtig zu erfahren, wie diese die Visite erleben. Dabei waren auch subjektive Mängel und Verbesserungsvorschläge von Interesse.

7.1.2 Usability-Test

Mithilfe von Usability-Tests soll die *Benutzbarkeit* (Usability) von Anwendungen und Webseiten untersucht werden. Dafür werden Standardaufgaben formuliert, die

alle Testpersonen ausführen sollen. Dabei ist zu beachten, dass es sich um ein für den Benutzer realistisches und relevantes Szenario handelt. Die Aufgabe sollte aus Sicht des Anwendungsgebietes formuliert sein und einen mittleren Schwierigkeitsgrad haben. Wichtig ist, dass die Aufgabenstellung nicht spezifisch auf die Applikation eingeht, sondern allgemeine Begriffe und Bezeichnungen verwendet. Vor jedem Test muss ein einheitlicher Systemzustand hergestellt werden, so dass alle Testpersonen die gleiche Umgebung vorfinden. Elementar bei der Durchführung ist, dass die Testperson gebeten wird, laut zu denken.

Es gibt nun zwei Möglichkeiten, den Test durchzuführen. Zum einen kann der Test formal ablaufen, d.h. der Beobachter greift nicht ein und bleibt möglichst passiv. Zum anderen kann der Testleiter den Ablauf moderieren, eingreifen und Fragen stellen. Dies wird als informeller Usability-Test bezeichnet.

MEDo wurde durch einen informellen Usability-Test evaluiert. Es wurde allerdings kein Versuchsaufbau in einem isolierten Usability-Labor gemacht. Stattdessen, um der Zeitknappheit der Ärzte Rechnung zu tragen, wurde der Test meist im Dienstzimmer auf der Station durchgeführt.

Bevor der Benutzer selber das System verwendete, wurde ihm eine kurze Einführung gegeben. Dabei wurde jeder Bildschirm kurz gezeigt und die Möglichkeiten erklärt. Dies dauerte im Schnitt ca. 3 Minuten.

Anhang A auf Seite 101 zeigt die Aufgabenstellungen der Ärzte. Dabei wurden mehrere Szenarien aus Abschnitt 4.4 auf Seite 35 kombiniert, um einen Durchschnitt durch die Aufgabenverwaltung zu bekommen. Die Aufgaben wurden jeweils nach Abschluss der vorangegangenen Aufgabe vom Testleiter gestellt, um die hohe Dynamik einer Visite abzubilden. Die Aufgaben sind in die zwei Phasen *Aufgabenerfassung* (während der Visite) und *Aufgaben bearbeiten* (nach der Visite) aufgeteilt. Nach Abschluss der zweiten Phase sollten möglichst alle Aufgaben als erledigt markiert sein.

7.1.3 Zweiter Teil des Interview

Nachdem der Benutzer Erfahrungen mit dem System sammeln konnte, wurde er zu seinen Eindrücken befragt. Wichtig war der erste Eindruck, der in weiteren Fragen detaillierter analysiert wurde. Hierzu wurde das Formblatt in Anhang A auf Seite 102 genutzt. Themen sind die Einsetzbarkeit, Eingabegeschwindigkeit, Einfluss auf die Kommunikation mit dem Patienten, Texteingabetyp und die wichtigsten Patienteninformationen. Am Ende der Umfrage wurden statistische Daten (Alter, Berufserfahrung und Geschlecht) aufgenommen und der Interviewpartner verabschiedet.

7.2 Ergebnisse

In den folgenden Abschnitten werden die Ergebnisse aus dem Usability-Test, den freien und den geschlossenen Fragen vorgestellt.

7.2.1 Erkenntnisse aus dem Usability-Test

Im Usability Test hat sich gezeigt, dass das System von allen Probanden sehr gut angenommen wurde. Wenn anfängliche Scheu vor dem Touchscreen vorhanden war, verschwand diese meist schon nach den ersten zwei bis drei Interaktionen.

Trotzdem haben sich bei der Beobachtung einige kleine Stolperfallen ergeben. Am interessantesten war sicherlich, dass die meisten Benutzer (90%) neue Untersuchungen über den entsprechenden Button im Diagnostikreiter (siehe Abbildung 7.2b) notiert haben. Nur ein Arzt nutzte die implizite Erkennung im „Neue Aufgabe“-Fenster (siehe Abbildung 7.2a) intuitiv. Zur Navigation zwischen den Patienten wurde immer der Stationsbildschirm genutzt, der Direktwahlbutton wurde genauso wie die Vor- und Zurückbuttons nicht genutzt.

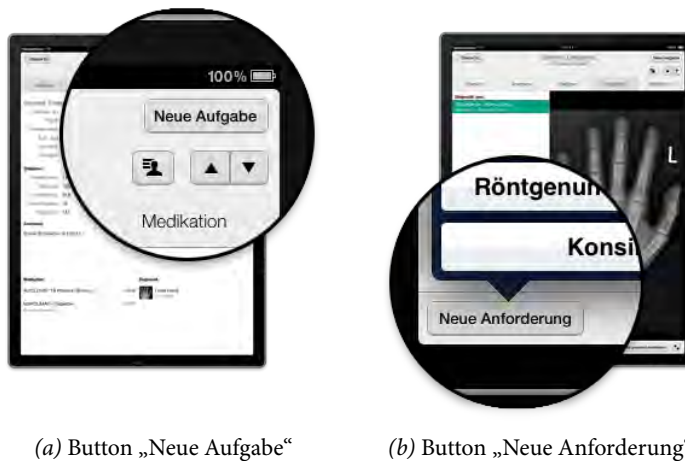


Abbildung 7.2: Vergleich der Positionen der Buttons um eine neue Anordnungs-Erinnerung zu erstellen

Es hat sich auch gezeigt, dass die Form des Buttons, der den Benutzer zurück zur Stationsübersicht bringt, sehr bei der Deutung der Funktion helfen kann: In der ersten Version – bei den ersten drei Interviews – hatte der Button keine Pfeilform (siehe Abbildung 7.3a). Hier musste den Benutzern Hilfe gegeben werden, um den Patientenbildschirm zu verlassen.

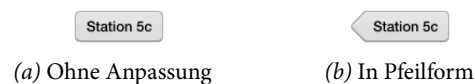


Abbildung 7.3: Zurück-Buttons

Sehr gut wurde die Anzeige des Status und die dahinterliegenden Prozesse angenommen. Verständnisprobleme sind hier nicht aufgetreten. Alle Ärzte haben sofort erkannt, dass das Konsil, nach dem gefragt wurde, noch nicht abgeschlossen ist. In

zwei Fällen entstand allerdings Verwirrung aus dem Umstand heraus, dass der Status im Imperativ („Patient untersuchen“) angezeigt wird.

7.2.2 Ergebnisse aus den geschlossenen Fragen

Im Durchschnitt bewerten die teilnehmenden Ärzte die Visite an ihrer Station als eher gut bis durchschnittlich, allerdings mit einer hohen Standardabweichung. Die Ergebnisse der Umfrage sind in Tabelle 7.1 aufgeführt.

Allgemein wurde das System als gut bewertet: Alle Ärzt/innen/e würden ein solches System auf der eigenen Station einsetzen und empfanden die Menüführung des Systems als logisch. Erstaunlicherweise evaluierte die Mehrheit die Eingabegeschwindigkeit des Systems als genauso schnell oder schneller als die Dokumentation mit Stift und Papier.

Tabelle 7.1: Ergebnisse der Befragung. Als Skala wurden die Schulnoten 1–6 verwendet, wobei Eins „Sehr gut“ und Sechs „Ungenügend“ entspricht

Frage	Mittelwert	Standardabw.
Aufgabennotation	2,22	1,30
Visite	2,50	1,12
Gesamteindruck	1,90	0,50
Sinnvoll auf Station	1,80	0,53
Eingabegeschwindigkeit	1,90	0,50
Patientenkommunikation	3,67	1,66
Menüführung	2,00	0,47

7.2.3 Ergebnisse des freien Teils des Interviews

In den freien Abschnitten des Interviews fanden sich viele weitere Meinungen zum System. Anwender, die mit dem SAP-System Erfahrung haben, fanden die textuelle Eingabe der Anforderungen eher umständlich. Von diesem System sind die Benutzer gewohnt, aus einer Hierarchie von Untersuchungen eine oder mehrere auszuwählen. Dies hat den Vorteil, dass die untersuchende Abteilung exakt weiss, was zu tun ist. Ansonsten besteht die Gefahr, dass nachgefragt werden muss, welche Untersuchung genau gewünscht wird oder es kommt zu fehlenden oder falschen Untersuchungen.

Im Vergleich mit Systemen mit Stylus¹ wurde der Touchscreen als besser verwendbar empfunden. Als ein Grund wurde angegeben, dass mit dem Stylus ein weiteres Objekt mitgeführt, bzw. verstaut werden muss, wenn die Hand für etwas anderes gebraucht wird.

Interessanterweise beobachtete Holzinger u. a. [29], dass Benutzer einen Stylus vorziehen und die Fehlerraten bei der Bedienung mit Fingern höher sind. Zwar wird dies in der Arbeit nicht erwähnt, aber es ist davon auszugehen, dass ein resistiver (auf

¹Stylus: Spitzer Stift, der zur Eingabe auf Touchscreens genutzt wird.

Druck reagierender) Touchscreen für die Fingererkennung eingesetzt wurde. Der in dieser Arbeit eingesetzte kapazitive (auf elektrische Leitfähigkeit des Körpers reagierende) Bildschirm könnte die Differenz erklären.

Viele Ärzte vermissten klar sichtbare Farbkodierungen für Statusangaben. Insbesondere war es für die befragten Chefarzte wichtig, nicht nur ein Rollensystem zu haben, sondern auch individuellen Personen Aufgaben zuzuweisen – so dass beispielsweise Studenten Blut abnehmen.

Gern gesehen wäre auch eine erweiterte Konfigurationsmöglichkeit der Textbausteine. Im Zusammenhang mit den bereits angesprochenen spezifischeren Anforderungen wurde vorgeschlagen, dass Unterkategorien bei Textbausteinen eingeführt werden. Durch diese könnten exakte Untersuchungen angegeben werden und so die Formulare ersetzt werden.

Weitere Ergebnisse finden sich im Folgenden:

Allgemein

- Viele Ärzte sind eine große Patientenübersicht oder Stationsliste gewöhnt, in der alle wichtigen Daten (Zimmernummer, Diagnose, anstehende Aufgaben, gepl. Entlassungstermin) zu allen Patienten aufgeführt sind (siehe Abbildung 7.4 auf der nächsten Seite).
- Weniger auf einmal anzeigen: Stationsbildschirm in mehrere Teilbildschirme aufteilen, jeweils Patienten-, Aufgaben- und bspw. Laborzentriert.
- Mehr Farbkodierungen, so dass dringende Aufgaben hervorgehoben werden.
- Aufgaben sollten filterbar sein: z.B. nur von heute.
- OP-Termine sollten angezeigt werden.
- Weitere, noch fehlende Patienteninformationen: Soziale Versorgung (Heim?), Kontaktdaten Verwandte, Hausarzt, Dokumente (alte Dokumente einscannen), Patient seit wann auf der Station und wie lange noch, OP Termine, Entlassungstermin, Prä-OP-Diagnose

Aufgaben/Anforderungen

- Anforderungen sollten schneller zu erreichen sein
- Viele Ärzte würden zwar sehr gerne Aufgaben per Sprache erstellen, aber nur, wenn diese fehlerfrei funktioniert.
- Die Formulare im System sind nicht als solche erkennbar, da die Textfelder keine Rahmen besitzen.
- Datumsangaben für Aufgaben wären für Benutzer von Interesse. Es wäre hier sowohl ein Fälligkeitsdatum als auch ein Start-Datum interessant.
- Prioritäten für Aufgaben

Diagnostika

- Im Normalfall wird ein Kumulativbefund eingesehen, keine einzelnen Laborergebnisse. Eine entsprechende Funktion, die alle bisherigen Untersuchungsergebnisse zusammenfasst, wird benötigt.
- Bei langliegenden Patienten kann eine große Menge von Untersuchungsergebnissen entstehen. Es sollten Filter angeboten werden, um nur einen bestimmten Typ oder nur Ergebnisse bis zu einem bestimmten Datum anzuzeigen.
- Nicht eingesehene Untersuchungsergebnisse sollten mehr hervorgehoben werden
- Auf chirurgischen Stationen (auch postchirurgische) werden bei einer Chefvisite viele Bilder hintereinander benötigt, die die Assistenzärzt/innen/e vorher aussuchen. Es wäre im System denkbar, bei der Anzeige eines Bildes eine Funktion bereitzustellen, um dieses einer Präsentation hinzuzufügen. Diese Präsentation kann dann während der Chefvisite angezeigt werden.
- Röntgenbilder sind oft nicht optimal belichtet. Für diesen Fall wäre ein Kontrastregler wünschenswert.

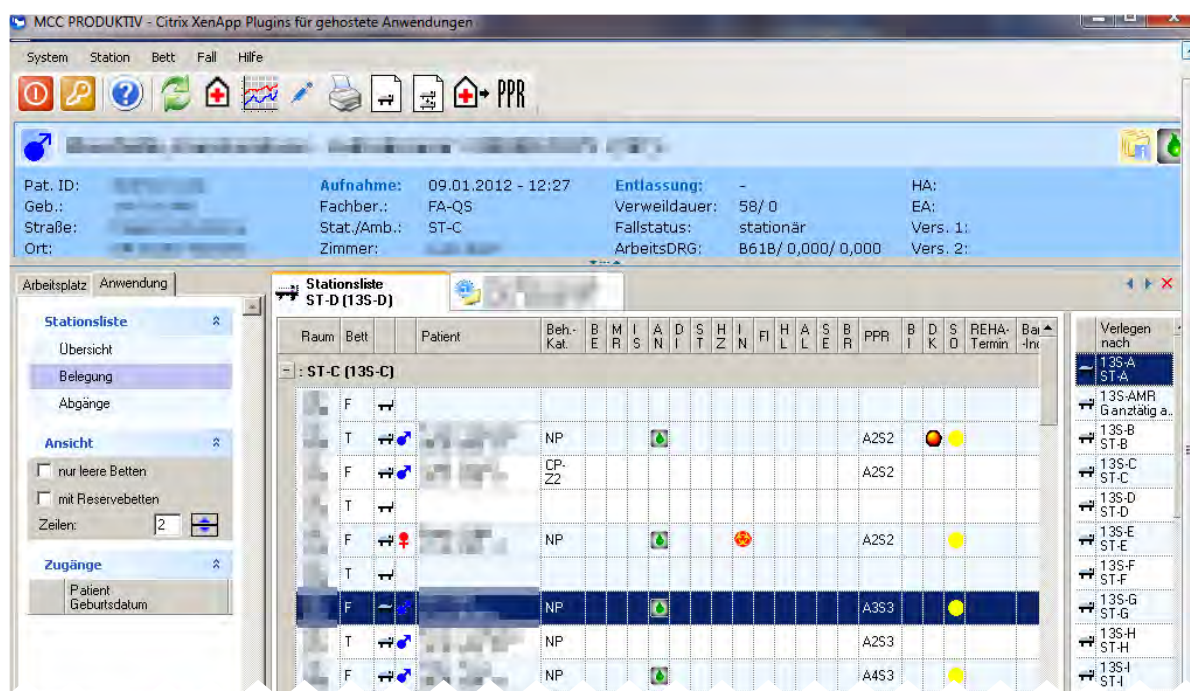


Abbildung 7.4: Stationsliste des MCC. Hier sind alle Patienten mit weiteren Details aufgeführt.

Medikation

- Medikation, Wechselwirkungen (Rote Liste) anzeigen. Medikation zeitlich begrenzen (mit entsprechendem Hinweis),
- Bedarfsmedikation mit Kurven korrelieren.

7.3 Diskussion

Die Evaluierung zeigte, dass auf Stationen, die mehrmals in der Woche Visite halten, eine mobile Unterstützung gerne angenommen wird. Besonders Ärzte, die bisher keine Möglichkeit haben, Patientendaten am Bett abzurufen, sind mit diesem Umstand unzufrieden. Interessanterweise waren gerade diese Ärzte der Meinung, dass ein System wie *MEDo* nicht die Kommunikation mit dem Patienten verbessern kann.

Insgesamt führte diese Frage – wie aus der Standardabweichung der Frage in Tabelle 7.1 auf Seite 86 zu sehen ist – zu sehr unterschiedlichen Antworten. Auf der einen Seite gab es diejenigen, die keinen Zusammenhang zwischen verwendeten Hilfsmitteln und der Kommunikation mit dem Patienten sehen. Die Begründung war hier oft, dass sich die eigentliche Kommunikation nicht ändert, und dass das Tablet nur die eigene Prozessgeschwindigkeit verbessert. Auf der anderen Seiten gab es die Meinung, dass es sehr wohl auf die Kommunikation Auswirkungen hat – beispielsweise dadurch, dass dem Patient am Bett Röntgenbilder gezeigt werden können.

Vielen gemein war die Ansicht, dass die Gefahr besteht, sich zu sehr mit dem Tablet bzw. den Informationen darauf zu beschäftigen – insbesondere wenn die Ärzt/innen/e alleine mit dem Tablet auf Visite sind. In diesem Fall würden viele versuchen, dass System nur während des Wechsels zwischen Patienten zu benutzen. Im Idealfall wünschen sich allerdings die meisten, dass die Visite mindestens zu zweit durchgeführt wird. So kann einer der Ärzt/innen/e das Gespräch leiten, während der andere das Gerät bedient.

8

Fazit

Die Arbeit begann mit der Beobachtung und Analyse von vier Visiten. Dazu wurden zum einen Prozessdiagramme zu den Visiten erstellt und verwendete Daten und Zeiten protokolliert. Die Prozessinformationen stellten die Grundlagen dar, um Gemeinsamkeiten und Unterschiede herauszuarbeiten. Aus den Gemeinsamkeiten wurde das Aufgabenmanagement als eine Chance herausgegriffen, um Prozessfehler, Laufzeiten und Unklarheiten bei der Visite zu vermindern.

Auf dieser Grundlage wurden Anforderungen für eine digitale Unterstützung der Visite mit einem Tablet erarbeitet, woraufhin ich ein Prototyp entwickelt habe, der diesen Anforderungen genügt. Mithilfe des Prototypen konnte durch Interviews und Usability-Tests gezeigt werden, dass eine Aufgabenverwaltung mit Prozessunterstützung der Arbeitsweise von Ärzten entgegenkommt und durch Verminderung von Brüchen in der Kommunikation Fehler, Wartezeiten und Wiederholungen verringern kann.

Insbesondere hat sich bei der Diskussion mit den Ärzten gezeigt, dass ein großes Interesse an einer besseren Aufgabenverwaltung und komplett integrierten KIS besteht. Auch der Vergleich mit den Benutzern des Motion-PC Tablets hat offensichtlich gemacht, dass Anwendungen speziell für Tablets entwickelt werden sollten, um eine bessere Nutzerakzeptanz zu erreichen.

8.1 Ausblick

Im Verlauf der Entwicklung sowie bei den abschliessenden Interviews haben sich einige Fragestellungen und Ideen ergeben, die in diesem Prototyp nicht weiter verfolgt werden konnten:

- Es sollte darauf verzichtet werden, mehrere Wege zur Erstellung von Aufgaben anzubieten. Dafür muss der „Anforderung erstellen“-Button entfernt und die Maske um Aufgaben zu erstellen verbessert werden. Beispielsweise könnten die Möglichkeiten der Textbausteine erweitert werden, so dass der Benutzer exakte Untersuchungen durch Untermenüs auswählen kann. Es sollte allerdings dabei darauf geachtet werden, dass schnelle, unspezifische Aufgaben weiterhin möglich sind.

- Bislang fehlt ein ausgearbeitetes Konzept, um Medikation und Vitaldaten anzuzeigen. Weiterhin sollte es möglich sein sein, Medikation und Vitaldaten in Aufgaben einzubinden. Beispielsweise könnten Medikamente für einen bestimmten Zeitraum verordnet werden. Rückt der Tag, an dem die Medikation ausläuft, näher, wird dem Benutzer die Aufgabe angezeigt, dass die Medikation zu überprüfen ist.
- Alle Maßnahmen im Krankenhaus müssen nach § 57 Abs. 1 BMV-Ä (Bundesmantelverträge für Ärzte und Zahnärzte) dokumentiert werden. Demnach soll die Dokumentation „die Anamnese, die Beschwerden des Patienten, die Diagnose und Behandlung sowie das Ergebnis [...] enthalten“ [48].

MEDo könnte dies auch unterstützen, in dem der „Als gesehen markieren“-Button durch einen Button, der dem Benutzer eine Dokumentationsmaske anbietet, ersetzt wird. In dieser kann er textuell oder per Diktat die Untersuchung befunden. Erst danach ist die Aufgabe erledigt.

- Es sollte weiterhin im Auge behalten werden, wie und wo die Daten gespeichert werden und wie auf diese zugegriffen werden kann. Die Homogenisierung der Systemlandschaft in Klinikbereich ist eine der großen Aufgaben der klinischen Informatik. Eine Möglichkeit bestünde darin, Adapter in Form von Webservices zu entwickeln.
- Weitere Probleme ergeben sich bei Stationen, bei denen Bilddaten eine große Rolle spielen. Besonders Schnittbilder, wie beispielsweise Bilder aus dem Computertomografen, führen dazu, dass bis zu 750 GB Netzwerktraffic pro Woche generiert werden. Bilddaten werden meist mithilfe von Digital Imaging Communication (DICOM) gespeichert.

DICOM ist ein nicht proprietäres Protokoll, Bildformat und Dateistruktur, um digitale Bilddaten im Klinikkontext anzuzeigen, übertragen und auszudrucken [10] und umfasst damit Geräte wie CT-Scanner und Radiografie, Workstations, Drucker, Archiv- und Bildverarbeitungssysteme.

Pro Patient können Datenmengen von bis zu 200 MB anfallen. Um Ladezeiten gering zu halten, sollten entweder aktuelle Daten auf dem Gerät zwischengespeichert oder ein Hochgeschwindigkeits-WLAN eingesetzt werden. Die Zwischenspeicherung hätte den weiteren Vorteil, dass das Gerät auch genutzt werden kann, wenn kein WLAN verfügbar ist. Es könnte dann beispielsweise alle Änderungen an Aufgaben zwischenspeichern und erst bei verfügbarer Verbindung Daten synchronisieren.

- Für die Anzeige von medizinischen Bildern existieren drei Klassen: Befundarbeitsplatz (Klasse A), Betrachtungsmonitor (Klasse B) und Sonstige (Klasse C). An einem Befundarbeitsplatz wird eine störungsfreie Umgebung geschaffen, beispielsweise durch einen hohen Maximalkontrast, hohe Leuchtdichte und 1:1 Zoom. An Betrachtungsmonitoren werden ähnliche, aber abgeschwächte Anforderungen gestellt. In die letzte Kategorie fallen alle anderen Geräte, wie Kontrollmonitore.

Die meisten Tablets fallen in Kategorie C, da eine Mindestpixelanzahl von 1000×1000 vorausgesetzt wird. Ersten Tests zufolge werden alle anderen Anforderungen erfüllt [7]. Neue Geräte, die einen ähnlich hohen DPI-Wert aufweisen wie manche aktuelle Smartphones (diese haben mehr als 300 DPI [63]), könnten dies ändern. Das neueste iPad (2012) hat beispielsweise eine Auflösung von 2048×1536 Pixeln.

8.2 Schlusswort

Im Verlauf dieser Arbeit wurde gezeigt, dass mithilfe von Tablets Daten, die ausschließlich digital vorliegen, nun mobil zugänglich sind und die Ärzt/innen/e bei der Visite unterstützen können. Dabei ist es durch die direkte Manipulation auch für neue Benutzer sehr einfach, die Software zu bedienen. Eine direkte, touch-optimierte Implementierung ist dabei sehr wichtig. Durch eine einfachere Desinfizierung und höhere DPI der Bildschirme würden sich weitere Einsatzzwecke von Tablets erschließen.

Es konnte auch gezeigt werden, dass die Kombination von Arbeitslisten und einfacher Aufgabenliste dabei hilft, von Komplexität durch Prozessmanagementsysteme zu abstrahieren. So stehen einfache Aufgaben neben von Prozessen vorgegebenen Aufgaben. Durch die Kombination von Prozessmanagement und mobiler, benutzerfreundlicher Plattform sind verteilte Aufgaben jederzeit erreichbar.

Anhang

A

Dokumente für die Datenerhebung

Im Folgenden die Dokumente, die zur Beobachtung der Visiten und für die Benutzerevaluation genutzt wurden.

Krankenhaus	Visite #
Daten <div style="border: 1px dashed black; height: 100px; width: 100%;"></div>	Personen <div style="border: 1px dashed black; height: 100px; width: 100%;"></div>

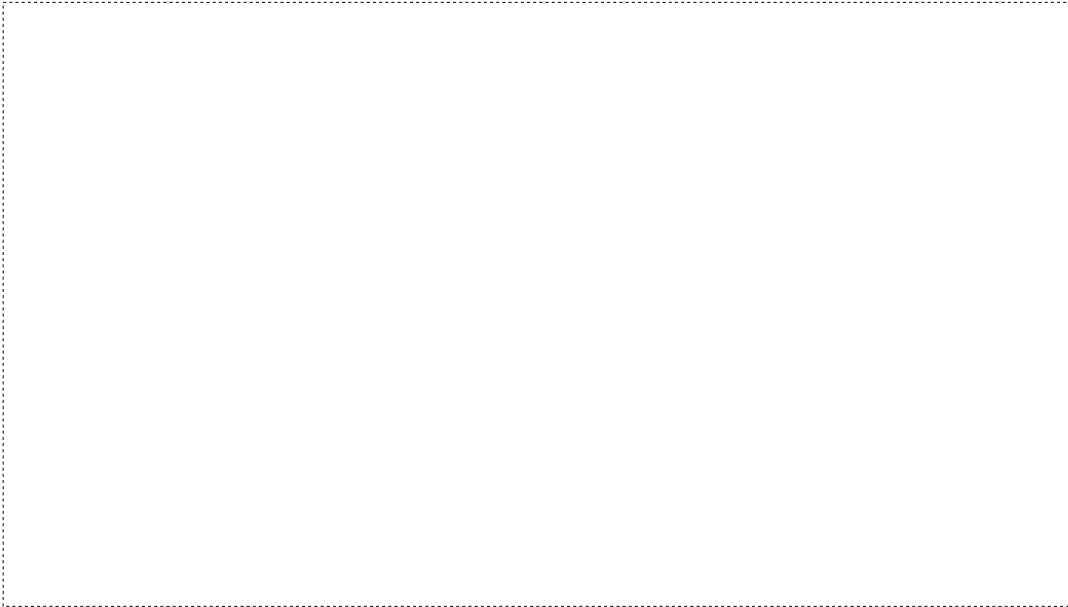
Phasen	Person	Tätigkeit	Sonderfall
Vorbereitung Beginn:			
Ante Portas Beginn:			
Am Krankenbett Beginn:			
Nachbereitung Beginn:			

Notizen <div style="border: 1px dashed black; height: 100px; width: 100%;"></div>	Eindruck <div style="border: 1px dashed black; height: 100px; width: 100%;"></div>
---	--

Ende

Krankenhaus

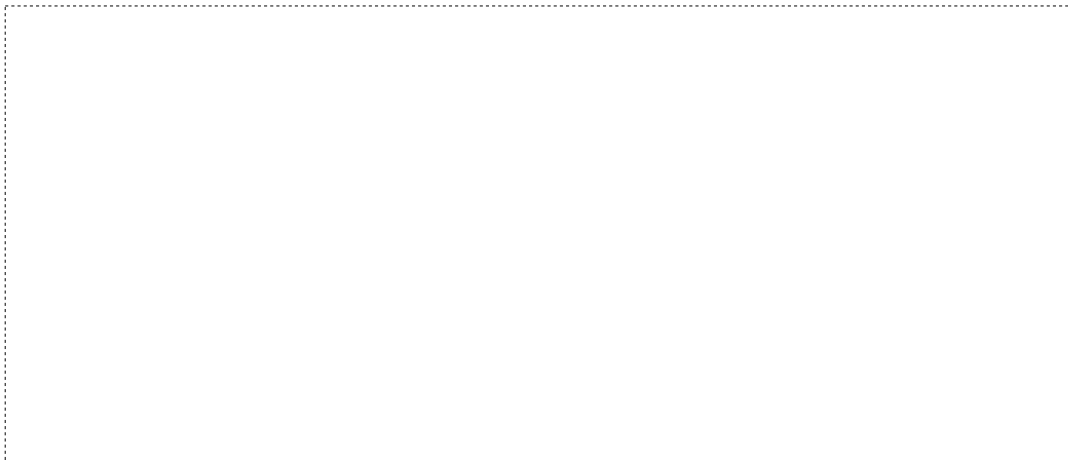
Daten



Technik



Notizen



Befragung

Vor dem Usability-Test

1. An welcher Station sind Sie tätig?
2. Wie wird die Visite an Ihrer Station ausgeführt?
3. (Wie wird mit dem Problem der digitalen Patientenakte umgegangen?)
4. Wie werden die Aufgaben, die während der Visite anfallen, notiert?
5. Bewerten Sie diese Aufgabennotation an Ihrer Station:
[1] [2] [3] [4] [5] [6]
6. Bewerten Sie die Visite an Ihrer Station im Allgemeinen:
[1] [2] [3] [4] [5] [6]
7. Welches sind die Hauptkritikpunkte an der Visite, wie Sie an Ihrer Station praktiziert wird?

Usability-Test

Spielregeln

- Testperson darf den Test jederzeit unterbrechen bzw. abbrechen
- Testperson sollte darum gebeten werden, laut zu denken
- Falls die Testperson mit einer Aufgabe nicht mehr weiterkommt, kann sie selbständig zur nächsten Aufgabe weitergehen.
- Der Beobachter sollte sich möglichst passiv verhalten

1. Während der Visite

Situation:

Sie befinden sich auf Visite am Bett von Patientin Crescentia Schmidt. Sie haben das Tablet mit dem MEDo-System bei sich. Auf diesem werden gerade die Daten von der Patientin, bei der Sie vorher waren angezeigt.

Aufgaben:

1. Sie wissen, dass für die Patientin Crescentia Schmidt neue Röntgenbilder eingetroffen sind. Zeigen Sie diese bitte an.
2. In der Diskussion ergibt sich, dass eine neue (große) Blutuntersuchung nötig ist. Notieren Sie sich diese Aufgabe.
3. Es kommen Fragen zur Medikation auf. Zeigen Sie diese an.
4. Es soll das vorherige Krankenhaus des Patienten angerufen werden, um die dortige Medikation zu erfahren. Erstellen Sie bitte eine neue Aufgabe.
5. Es geht weiter zur nächsten Patientin: Kethe Pakuscher. Für diese Patientin war ein Konsil angefordert worden. Schauen Sie nach, ob dieses bereits eingetroffen ist.
6. Fordern Sie eine neue Röntgenaufnahme des rechten Knies an. Erstellen Sie dazu eine neue Aufgabe.

Die Visite sei hiermit nun beendet.

2. Nach der Visite

Aufgaben:

1. Bearbeiten Sie die neu angefallenen Aufgaben entsprechend (drei an der Zahl).
Wenn neue Diagnostik eintrifft, markieren Sie diese als gesehen.
Sie sollten keine unbearbeitete Aufgaben mehr haben zur Abschluß dieser Aufgabe.

Befragung

Nach dem Usability-Test

1. Gesamt-Eindruck des vorgestellten Systems
[1] [2] [3] [4] [5] [6]
2. Ist ein solches System auf Ihrer Station vorstellbar/sinnvoll?
[1] [2] [3] [4] [5] [6]
3. Kann das System von der Eingabegeschwindigkeit her mit dem Notiz-System auf der Station mithalten?
[1] [2] [3] [4] [5] [6]
4. Denken Sie, dass ein solches System die Kommunikation mit den Patienten während der Visite verbessern kann?
[1] [2] [3] [4] [5] [6]
5. Empfinden Sie die Menüführung als sinnvoll?
[1] [2] [3] [4] [5] [6]
6. Was vermissen Sie spontan?
7. Was fanden Sie grafisch ansprechend / Was nicht?
8. Welche Methode der Texteingabe finden Sie/bzw fanden Sie am sinnvollsten: Text per Tastatur, Textbausteine oder Audioaufnahme?
9. Würden Sie gerne mehr Kontrolle über die Anordnung der Inhaltsblöcke haben?
10. Was sind außerhalb des Aufgabenmanagements die wichtigsten Patienteninformation für Sie (Medikation, Anamnese, etc)
11. Statistische Daten:
 - a. Alter
 - b. Berufserfahrung (Jahre)?
 - c. Geschlecht

Vielen Dank!

Literaturverzeichnis

- [1] AALST, W.M.P. Van d. u. a.: The application of Petri nets to workflow management. In: *Journal of Circuits Systems and Computers* 8 (1998), S. 21–66
- [2] ADAM, N.R. ; ATLURI, V. ; HUANG, W.K.: Modeling and analysis of workflows using Petri nets. In: *Journal of Intelligent Information Systems* 10 (1998), Nr. 2, S. 131–158
- [3] AMBER, Scott W.: *Introduction to User Interface (UI) Prototypes*. Version: 2009. <http://www.agilemodeling.com/artifacts/uiPrototype.htm>, Abruf: 13 Februar 2012
- [4] AMMENWERTH, E. ; BUCHAUER, A. ; BLUDAU, B. ; HAUX, R.: Mobile information and communication tools in the hospital. In: *International journal of medical informatics* 57 (2000), Nr. 1, S. 21–40
- [5] AMMENWERTH, E. ; MANSMANN, U. ; MAHLER, C. ; KANDERT, M. ; EICHSTADTER, R.: Are quantitative methods sufficient to show why wards react differently to computer-based nursing documentation? In: *Studies in health technology and informatics* (2002), S. 377–381
- [6] APPLE INC. (Hrsg.): *iOS Human Interface Guidelines*. Version: 10 2011. <http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html>, Abruf: 22. Februar 2012
- [7] AYCAN DIGITALSYSTEME GMBH (Hrsg.): *iPad: aycan führt erfolgreiche Abnahmeprüfung nach DIN V 6868-57 durch*. <http://aycandigital.blogspot.com/2010/04/ipad-aycan-fuhrt-erfolgreiche.html>, Abruf: 19. Februar 2012
- [8] BALLAGAS, R. ; BORCHERS, J. ; ROHS, M. ; SHERIDAN, J.G.: The smart phone: a ubiquitous input device. In: *Pervasive Computing, IEEE* 5 (2006), Nr. 1, S. 70–77
- [9] BÄUMER, Dirk ; BISCHOFBERGER, Walter R. ; LICHTER, Horst ; ZÜLLIGHOVEN, Heinz: User interface prototyping – concepts, tools, and experience. In: *Proceedings of the 18th international conference on Software engineering*. Washington,

- DC, USA : IEEE Computer Society, 1996 (ICSE '96). – ISBN 0-8186-7246-3, 532-541
- [10] BIDGOOD JR, W.D. ; HORII, S.C. ; PRIOR, F.W. ; VAN SYCKLE, D.E.: Understanding and using DICOM, the data interchange standard for biomedical imaging. In: *Journal of the American Medical Informatics Association* 4 (1997), Nr. 3, S. 199-212
- [11] BLUM, K. ; MÜLLER, U.: *Dokumentationsaufwand im ärztlichen Dienst der Krankenhäuser*. Dt. Krankenhaus-Verl.-Ges., 2003
- [12] COOPER, A. ; REIMANN, R. ; CRONIN, D.: *About Face: Interface und Interaction Design*. Hüthig Jehle Rehm, 2010
- [13] DADAM, Peter ; REICHERT, Manfred ; KUHN, Klaus: Clinical Workflows - The Killer Application for Process-oriented Information Systems? In: *Proc. 4th Int'l Conf. on Business Information Systems (BIS'00)*, Springer, April 2000, S. 36-59
- [14] DEMMEL, Hans-Joachim ; ELIGEHAUSEN, Sigmar ; GEISLER, Linus S. ; HANNICH, Hans-Joachim ; HOEFERT, Hans-Wolfgang ; RIPPmann, Konrad: *Kommunikation als Erfolgsfaktor im Krankenhaus*. Bd. 1. medhochzwei Verlag, 2008
- [15] DIERKESMANN, Rainer ; WALDMANN, Werner: Ärztliche Visite im Krankenhaus - Lernen aus der Vergangenheit? In: *Kliniker* 39 (2010), Nr. 07/08. <http://dx.doi.org/10.1055/s-0030-1265815>. – DOI 10.1055/s-0030-1265815. ISBN 0341-2350
- [16] DIETZEL, G.T.W.: E-Health für eine bessere Gesundheitsversorgung in Deutschland. In: *Telemedizinführer Deutschland* (2005), S. 22-26
- [17] Norm DIN 66001 September 1966. *Sinnbilder für Datenfluss- und Programmablaufpläne*
- [18] DPA: Mit dem iPad zur Visite. In: *Gesundheit und mehr... (Patientenmagazin des Universitätsklinikum Leipzig)* 22 (2011)
- [19] ENENKIEL, Kerstin: *healthHistory - Konzeption & Implementierung einer mobilen Patientenakte*, Universität Ulm, Diplomarbeit, Mai 2011. <http://dbis.eprints.uni-ulm.de/748/>
- [20] FRÄNKEL, Katrin: *Evaluation des Nutzenpotentials mobiler Dokumentationswerkzeuge zur Unterstützung der klinischen Dokumentation im Krankenhaus*, Universität Erlangen-Nürnberg, Dissertation, 2010
- [21] FRASCH, Eugen: *Multimedia-Inhalte im Android Framework - eine Erklärung anhand einer App*, Universität Ulm, Bachelorarbeit, Januar 2012. <http://dbis.eprints.uni-ulm.de/794/>

- [22] FREUND, Jakob ; GÖTZER, Klaus: *Vom Geschäftsprozess zum Workflow*. Carl Hanser Verlag GmbH & CO. KG, 2008
- [23] GADATSCH, A. (Hrsg.) ; NÜTTGENS, M. (Hrsg.) ; RUMP, F. J. (Hrsg.) ; GI-Arbeitskreis „Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)“ der GI-Fachgruppe WI-MobIS (FB-WI) (Veranst.): *Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*. Bonn : Gesellschaft für Informatik, November 2007 (6)
- [24] GEORGAKOPOULOS, D. ; HORNICK, M. ; SHETH, A.: An overview of workflow management: From process modeling to workflow automation infrastructure. In: *Distributed and parallel Databases* 3 (1995), Nr. 2, S. 119–153
- [25] GESUNDHEITSMINISTERIUM FÜR GESUNDHEIT (Hrsg.): *Die elektronische Gesundheitskarte*. Version: September 2011. http://www.bmg.bund.de/fileadmin/dateien/Publikationen/Infoblaetter/GP_Infoblatt_10_2011.pdf, Abruf: 20. März 2012. GP_Infoblatt
- [26] GÖRES, U.: Nutzerakzeptanz – Herausforderung Telemedizin am Beispiel der elektronischen Gesundheitskarte. In: *Telemedizinführer Deutschland* (2009), S. 272–280
- [27] GREENES, R.A.: *Clinical decision support: the road ahead*. Academic Press, 2007
- [28] GRÜNDLER, M. ; SPEKKER, H. ; NEE, O. ; EICHELBERG, M. ; KRONBERG, K.: M³IS Online-Zugriff auf multimediale Patientendaten mit mobilen Endgeräten am Krankenbett / Kuratorium OFFIS eV, Klinikum Oldenburg gGmbH. – Forschungsbericht
- [29] HOLZINGER, A. ; HOLLER, M. ; SCHEDLBAUER, M. ; URLESBERGER, B.: An investigation of finger versus stylus input in medical scenarios. In: *Information Technology Interfaces, 2008, 30th International Conference on*, 2008. – ISSN 1330–1012, S. 433 –438
- [30] KAY, Alan C.: *A Personal Computer for Children of All Ages*. Xerox Palo Alto Research Center, 1972. – Draft
- [31] KOHL, Andreas: *Erstellung und Einsatz von Prototypen*. Version: Dezember 2010. http://www.ef.eu/wp-content/uploads/2010/12/efeu_usability_workshop_kohl_101215_handout.pdf, Abruf: 21. März 2012. Usability Workshop
- [32] KREOWSKI, H.J.: Formale Sprachen: Graphtransformation. In: *Universität Bremen, Sommersemester* (2004)

- [33] LANSDALE, MW: On the memorability of icons in an information retrieval task. In: *Behaviour & Information Technology* 7 (1988), Nr. 2, S. 131–151
- [34] LENZ, Richard ; PELEG, Mor ; REICHERT, Manfred: Healthcare Process Support: Achievements, Challenges, Current Research. In: *International Journal of Knowledge-Based Organizations (IJKBO)* (2012). <http://dbis.eprints.uni-ulm.de/784/>
- [35] MALE, M. D.: *Toshiba Portege M200*. Version: 2010. <http://www.flickr.com/photos/mastermaq/4836376265/>, Abruf: 21. März 2012
- [36] MENNFELD: *Dokumappen für Pflegende im Krankenhaus*. Version: November 2005. <http://www.pflegewiki.de/wiki/Datei:Dokumappe.JPG>, Abruf: 16. März 2012. Vervielfältigung unter Beachtung der GNU Free Documentation License, Version 1.2
- [37] MINER, Wilson: *When We Build*. Version: November 2011. <http://wm4.wilsonminer.com/build2011/>, Abruf: 17. März 2012
- [38] NEUBAUER, A. S. ; PRIGLINGER, S. ; EHRT, O.: Elektronische oder papiergebundene Patientenakte Ein Kosten-Nutzen-Vergleich. In: *Der Ophthalmologe* 98 (2001), 1083–1088. <http://dx.doi.org/10.1007/s003470170030>. – ISSN 0941–293X. – 10.1007/s003470170030
- [39] NIELSEN, Jakob: *Usability Engineering*. Morgan Kaufmann, 1993
- [40] NÜTTGENS, M. ; RUMP, F.J.: Syntax und semantik ereignisgesteuerter prozessketten (epk). In: *Promise* Bd. 2, 2002, S. 64–77
- [41] OUYANG, C. ; VAN DER AALST, W.M.P. ; DUMAS, M. ; TER HOFSTEDE, A.H.M.: *Translating BPMN to BPEL*. (2006)
- [42] PETRI, C.A.: *Communication with automata: Volume 1 Supplement 1*, Technische Hochschule Darmstadt, Dissertation, November 1966
- [43] RECKER, J.C. ; INDULSKA, M. ; ROSEMAN, M. ; GREEN, P.: How good is BPMN really? Insights from theory and practice. (2006)
- [44] REICHERT, M.: Prozessmanagement im Krankenhaus: Nutzen, Anforderungen und Visionen. In: *das Krankenhaus* 92 (2000), Nr. 11, S. 903–909
- [45] REICHERT, Manfred: *Dynamische Ablaufänderungen in Workflow-Management-Systemen*, Universität Ulm, Dissertation, Juli 2000. <http://dbis.eprints.uni-ulm.de/433/>

- [46] REICHERT, Manfred ; RINDERLE-MA, Stefanie ; DADAM, Peter: Flexibility in Process-aware Information Systems. In: *LNCS Transactions on Petri Nets and Other Models of Concurrency (ToPNoC), Special Issue on Concurrency in Process-aware Information Systems*. 2 (2009), März, 115–135. <http://dbis.eprints.uni-ulm.de/481/>
- [47] ROBECKE, Andreas: *Development of an iPhone business application*, Universität Ulm, Diplomarbeit, Februar 2011. <http://dbis.eprints.uni-ulm.de/716/>
- [48] SCHIRMER, Horst D. ; HÜBNER, Marlies: Ärztliche Dokumentationspflichten: Das Ende der Fahnenstange. In: *Deutsches Ärzteblatt* 106 (2009), November, Nr. 48, S. A–2208. – Online verfügbar unter <http://www.aerzteblatt.de/archiv/66915>
- [49] SCHMID, Maximilian: *Mobile Web Service Implementierung am Beispiel des iPhone OS 4.0*, Universität Ulm, Bachelorarbeit, November 2010. <http://dbis.eprints.uni-ulm.de/709/>
- [50] SCHUMM, D.: *Graphische Modellierung von BPEL Prozessen unter Verwendung der BPMN Notation*, Universität Stuttgart, Diplomarbeit, 2008
- [51] SKOENTZKI-CHENG, S.: *Die elektronische Fieberkurve zur Mobilen Visite und Dokumentation im Krankenhaus*. Version: 05 2010. <http://www.ssk.med.uni-erlangen.de/de/vortraege/56-elektronische-fieberkurve/133-2010-zwickau-die-elektronische-fieberkurve-vmobile>, Abruf: 21. März 2012. Präsentation
- [52] STAUSBERG, J. ; USLU, A. ; SCHOCH, B.: Die Elektronische Patientenakte in der Intensivmedizin: Anforderungen-Konzepte-Nutzen. In: *Telemedizinführer Deutschland*, Minerva, Darmstadt (2004), S. 136–140
- [53] THOMPSON, AG ; JACOB, K. ; FULTON, J. ; MCGAVIN, CR: Do post-take ward round proformas improve communication and influence quality of patient care? In: *Postgraduate medical journal* 80 (2004), Nr. 949, S. 675
- [54] TIDWELL, J.: *Designing interfaces*. O'Reilly Media, Inc., 2005
- [55] VIRZI, R.A. ; SOKOLOV, J.L. ; KARIS, D.: Usability problem identification using both low-and high-fidelity prototypes. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground* ACM, 1996, S. 236–243
- [56] VOGT, Maren: *Visite als Planungs- und Steuerungsinstrument in der Pflege und Therapie im Krankenhaus: arbeitspsychologische Studie auf zwei Stationen der Inneren Medizin*. Verlag Dr. Kovac, Hamburg, 2003. – S. 15, S. 26, S. 36–40, S. 352–358, S. 372–375

- [57] WALKER, M. ; TAKAYAMA, L. ; LANDAY, J.A.: High-fidelity or low-fidelity, paper or computer? Choosing attributes when testing web prototypes. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* Bd. 46 SAGE Publications, 2002, S. 661–665
- [58] WALTER, M. ; PUHL, H. ; EKKERNKAMP, A.: meditrace: Zeitersparnis und Qualitätsverbesserung durch standardisierte, mobile Befunddokumentation. In: „*Mobiles Computing in der Medizin*“, *Proceedings zum 1* (2001), S. 114–127
- [59] WEBER, H. ; STÖCKLI, M. ; NÜBLING, M. ; LANGEWITZ, W. A.: Communication during ward rounds in Internal Medicine: An analysis of patient–nurse–physician interactions using RIAS. In: *Patient education and counseling* 67 (2007), 08, Nr. 3, 343–348. <http://linkinghub.elsevier.com/retrieve/pii/S0738399107001553?showall=true>. ISBN 0738–3991
- [60] WEISER, M.: The computer for the 21st century. In: *Scientific American* 265 (1991), Nr. 3, S. 94–104. – Online verfügbar unter <http://sandbox.xerox.com/want/papers/ubi-sciam-sep91.pdf>
- [61] WHITE, S.A.: Introduction to BPMN / IBM Cooperation. 2004. – Forschungsbericht
- [62] WIEDENBECK, S.: The use of icons and labels in an end user application program: an empirical study of learning and retention. In: *Behaviour & Information Technology* 18 (1999), Nr. 2, S. 68–82
- [63] WIKIPEDIA (Hrsg.): *List of displays by pixel density*. Version: März 2012. http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density, Abruf: 14. März 2012

C

Akronyme

API Application Programming Interface.
BPMN Business Process Modeling Notation.
DICOM Digital Imaging Communication.
DPI Dots per Inch.
DSS Decision Support System.
EGK elektronische Gesundheitskarte.
EPA elektronische Patientenakte.
EPK Ereignisgesteuerte Prozessketten.
GCD Grand Central Dispatch.
IDE Integrated Development Environment.
KIS Krankenhausinformationssystem.
MCC Meierhofer Clinical Competence.
MVC Model View Controller.
OMG Object Management Group.
PDMS Patientendatenbankmanagementsystem.
PPI Pixel per Inch.
RKU Universitäts- und Rehabilitationskliniken Ulm.
SDK Software Development Kit.
WfMS Workflow-Management-System.
WLAN Wireless Local Area Network.

Abbildungsverzeichnis

1.1	Übersicht über den Aufbau der Arbeit	3
1.2	UI prototyping process	4
2.1	Mappen für die Pflegedokumentation	6
2.2	Ein Prozessbeispiel in BPM-Notation	8
2.3	<i>Process mangagement lifecycle</i>	9
2.4	Applikation Erinnerungen	12
2.5	Beispiele für Tablet Computer	13
2.6	Vision des ubiquitären Computers	14
2.7	Wahrnehmbare Teile des Computers und deren Entwicklung in Richtung von Tablets	15
3.1	Prozessmodell Visite in der Inneren Medizin	20
3.2	Prozessmodell Visite in der Notaufnahme	22
3.3	Mobile Clinical Assistant von Motion Computing, Inc.	23
3.5	Bildschirmatatur und Stift des Mobile Clinical Assistant	23
3.4	Prozessmodell Visite in der Orthopädie	24
3.6	Prozessmodell Visite in der Chirurgie	26
3.7	Prozessmodell Röntgenuntersuchung	27
3.8	Prozessmodell Anforderung einer Blutuntersuchung	27
4.1	Übersicht über die Phasen bis zur Evaluierung des Prototypes	30
4.2	Patientenakte im MCC. Um zum Kumulativbefund zu gelangen, muss der Benutzer den Patienten auswählen, in der rechten Leiste „Arbeitsplatz“ anklicken und in einem Dropdown „Kumulativbe- fund“ auswählen.	33
4.3	Modell der Patientendaten	34
5.1	Unterscheidung zwischen horizontalen und vertikalen Prototypen .	37
5.2	Informationsarchitektur	39
5.3	Projektion von Inhalten auf eine zweidimensionale Ebene	40
5.4	Implementierungs-, Repräsentations- und Usermodell	41
5.5	Repräsentationsmodelle für die Navigation zwischen Patienten und deren Daten	41

5.6	Verschiedene Aufteilungen der Stationsübersicht	42
5.7	Entwürfe zu Aufgabenfenstern	43
5.8	Navigationsleisten	45
5.9	Entwürfe zur Darstellung der Patientendaten	46
5.10	Patientendaten in Portraitorientierung im Raster, mit Tab-Leiste . .	47
5.11	Tab-Leiste am oberen Rand	48
5.12	Entwurf eines Buttons	48
5.13	Verschiedene Pixeldesigns der Buttons	48
5.14	Raster	49
5.15	Farben	49
5.16	Entwürfe für verschiedene Icons	51
6.1	Model View Controller	54
6.2	Datenbankaufbau	55
6.3	Angepasste Interface-Elemente	57
6.4	MDTextModuleListControl nach Antippen des Buttons „Medikation“	58
6.5	Berechnung der Koordinaten zum Einfügen des Textes	59
6.6	Storyboardausschnitt	60
6.7	Start-Bildschirm	61
6.8	Login-Bildschirm	62
6.9	Stationsübersicht	64
6.10	Patientenbildschirm	66
6.11	Patientenbildschirm – Diagnostika	68
6.12	Patientenbildschirm – Laborergebnisse	70
6.13	Patientenbildschirm – Laboruntersuchung anfordern	70
6.14	Patientenbildschirm – Medikation	71
6.15	Patientenbildschirm – Vitaldaten	71
6.16	Bildschirm: Neue Aufgabe erstellen	72
6.17	Bildschirm: Aufgabedetails anzeigen	74
6.18	Darstellung des normalen UISplitView in verschiedenen Orientie- rungen	75
7.1	Ablauf des Evaluierungsprozesses	83
7.2	Vergleich der Positionen der Buttons um eine neue Anordnungs- Erinnerung zu erstellen	85
7.3	Zurück-Buttons	85
7.4	Stationsliste des MCC	88

E

Tabellenverzeichnis

3.1	Allgemeine Informationen über die betrachteten Stationen	19
4.1	Zugriffe auf Patientendaten	35
4.2	Szenarien	36
6.1	Abgleich der Anforderungen mit Implementierung	80
7.1	Ergebnisse der Befragung	86

Name: David Langer

Matrikelnummer: 622552

Erklärung

Ich erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

David Langer